

UNIVERSITY OF PENNSYLVANIA

THE WHARTON SCHOOL

OPIM 101 PKG.1

Professor Kimbrough/Laing

Spring 1996

Table of Contents

1. Wharton Computing Notes: DOS & UNIX Files, elm, editor, and Tin
2. Computerworld, Business Week, Money and Wall Street Journal Articles
3. "The Hypertext Markup Language," Chapter 4 from Spinning the Web, by Andrew Ford
4. "Notes for: An Introduction to Decision Technologies," by Kimbrough, Laing, and Lohse, chapters 3
5. "Judgment under Uncertainty," chapter 2 of Judgment in Managerial Decision Making, 2nd ed., By Bazerman
6. "A Gentle Introduction to Genetic Algorithms," chapter 1 of Genetic Algorithms in Search, Optimization, and Machine Learning, by Goldberg
7. Internet article, Technology Review, May/June 1995, pp.24-31
8. Information retrieval article, "An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System," Blair and Maron, Communications of the ACM, 1985, 28(3), pp.289-299
9. Neural network article, Communications of the ACM, 1994, 37(3), 93-105
10. Genetic algorithms article, Communications of the ACM, 1994, 37(3), 113-119
11. Crystal Ball version 3.0 User Manual
12. Braincel version 2.0 Tutorial
13. Using Evolver--A Tutorial
14. Homework exercises
15. Lecture notes/slides
16. Fall 1995 final examination

UNIVERSITY OF PENNSYLVANIA

THE WHARTON SCHOOL

OPIN 101 PKG1

Professor Kimbrough, ailing

Spring 1996

Table of Contents

1	Wharton Computing Notes: DOS & UNIX Files, 2nd edition and 1st
2	Computational Business Week: Money and Wall Street Journal Articles
3	"The Hypertext Markup Language," Chapter 4 from Spinning the Web, by Andrew Ford
4	"Notes for: An Introduction to Decision Technologies," by Kimbrough, Loring, and Lohar, chapters 1-3
5	"Judgment under Uncertainty," chapter 2 of Judgment in Managerial Decision Making, 2nd ed., by Baruch
6	"A Gentle Introduction to Genetic Algorithms," chapter 1 of Genetic Algorithms in Search, Optimization, and Machine Learning, by Goldberg
7	Internet article: Technology Review, May/June 1995, pp. 34-41
8	Information retrieval article: "An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System," Helt and Nelson, Communications of the ACM, 1985, 28(3), pp. 280-290
9	Neural network article: Communications of the ACM, 1991, 34(5), 98-108
10	Genetic algorithm article: Communications of the ACM, 1991, 34(5), 113-119
11	Crystal Ball version 3.0 User Manual
12	Biological version 3.0 Tutorial
13	Using Evolver-A Tutorial
14	Homework exercises
15	Lecture notes/slides
16	Fall 1995 final examination

and in 11-14-1995, and the following lead in the report (see "Connecting from the DOS/Windows Computer Labs")

"If you have a modem, communications software, and a telephone network ID and password, you can also access Wharton's host systems by dialing in to PennNet over telephone lines from off-campus, or from the Sun Lodge, Laptop Computer Bar, in James Hall (see "Connecting by Dialing in")."

"Many offices can access host systems through a connection to Wharton's network (see "Connecting from Wharton Offices")."

"Once you connect to the system, you need to log on, typically by entering your account name (or 'username') and then your password (see "Log the Log"). On some systems you are required to change your password when you log on at a given time (see "Changing Your Password")."

Logging On Wharton's Host Systems

The Wharton School has several "host" computer systems available for remote access, including Wharton's Unix systems (which include hosts Futures, Equity, and Assets), and Wharton's Academic VMS System (host Wilma).

Before logging on to a Wharton host system, you need an account on that system. Accounts for most Wharton systems are available from Wharton's Accounts Coordinator in 212 Vance Hall. For more information, contact the Accounts Coordinator at 898-0750.

If you plan to connect to a host system using PennNet or PennNet's dial-in lines, you also need a PennNet network ID and password. With a valid PennCard, you can get a PennNet ID and password from several locations including Wharton's Accounts Coordinator in 212 VH. For questions or more information, contact Wharton's computer consultants at 898-8600 or the PennNet Services Center at 898-8171.

Once you have an account on the host system, to use the system you need to (1) connect to the system and then (2) log on as described the following sections.

Access to Wharton's host systems is available over Wharton's Ethernet network or PennNet through several methods:

- ▶ You can access most Wharton host systems from the microcomputers in Wharton's DOS/Windows computer labs, including the MBA Lab in 210-11 VH, the DOS/Windows Lab in 114 SH-DH, and the Training Lab in 116 SH-DH. (see "Connecting from the DOS/Windows Computer Labs").
- ▶ If you have a modem, communications software, and a PennNet network ID and password, you can also access Wharton's host systems by dialing in to PennNet over telephone lines from off-campus or from the Sun Lounge "Laptop Computer Bar" in Vance Hall (see "Connecting by Dialing In").
- ▶ Many offices can access host systems through a connection to Wharton's network (see "Connecting from Wharton Offices").

Once you connect to the system, you need to log on, typically by entering your account name (or "username") and then your password (see "Logging On"). On some systems you are also required to change your password when you log on for the first time (see "Changing Your Password").

Connecting from the DOS/Windows Computer Labs

The DOS/Windows computer labs in Steinberg Hall-Dietrich Hall and Vance Hall offer high-speed Ethernet connections to Wharton's network. To connect to a Wharton host system from a lab computer, do the following:

Step 1 Start the lab computer and go into Windows as follows:

- ▶ If the lab computer is off, turn on the computer and, if necessary, the monitor. (The monitor may take several seconds to warm up.)
- ▶ If the computer is already on, simultaneously press the Ctrl, Alt, and Del keys to reboot the computer.
- ▶ When the screen prompts you to "Enter your user ID:," enter your last name. (You don't need to enter a password.)

It is important to reboot the computer and enter your own "user ID" each time you use a lab computer. If you need to reboot the computer again while you're using it, enter the same user ID. This prevents the computer from performing a routine purge of the hard disk and erasing your files.

Once you enter your user ID, you will see the lab's Main Menu.

- ▶ From the Main Menu, type `win` to start Microsoft Windows.

Step 2 Open a telnet session to the host system as follows:



Equity

- ▶ Several commonly-used Wharton host systems—such as Unix systems Equity, Futures, and Assets, and VMS system Wilma—have their own icon in the "Network Host Access" group.
 - If the host you want is listed, simply double-click its icon.



Telnet

- ▶ If you want to connect to another host system, or a system outside Wharton, do the following:

- Double-click on the "Telnet" icon.

This displays an "Open Session" dialog box.

- Type the name of the system in the Hostname box.

For a system outside Wharton, enter the full Internet name of the host (for example `archie.rutgers.edu`).

You can leave the Session Name blank.

- Click the "OK" button.

If you've entered a valid host name, you'll see a "Connecting to host" message as you connect to the system. Depending on the system, you may see a brief message identifying the system and then be prompted for your user name (see "Logging On"). When you are finished, be sure to log off the system (see "Logging Off").

Connecting by Dialing In

If you have a modem and communications software (such as MS-Kermit, ProComm, or MicroPhone II) you can connect to Wharton's host systems over telephone lines by dialing in to PennNet, the University's communications network. To use PennNet, you also need a PennNet network ID and password (see page 1).

To dial in through PennNet:

Step 1 Set up your modem and configure your communications software.

- ▶ The communications settings for PennNet are 8 data bits and no parity. PennNet supports transfer rates of 1200, 2400, 9600, and 14,400 bps.
- ▶ Select VT100 or VT102 terminal emulation.

Refer to the instructions for your modem and communications software for more information.

Step 2 Start your communications software and dial PennNet at 898-0834.

- ▶ From an on-campus phone—including the phone lines at Wharton's Laptop Computer Bar—simply dial 8-0834.

Most communications packages offer automatic dialing—you simply need to tell the program the phone number. If your software doesn't offer automatic dialing, you need to enter terminal emulation, type in a "dialing prefix" code and then the phone number. On most modems, the dialing prefix for a touch-tone line is **ATDT**, and for a pulse or rotary line is **ATDP**.

For example, to dial PennNet from an off-campus phone using a touch-tone telephone line, enter **ATDT8980834**.

Step 3 Once your modem connects to PennNet, press the Enter key a few times.

PennNet responds with a connect-message that begins:

Annex Command Line Interpreter

Step 4 If necessary, press the Enter key once more until PennNet displays several lines of information and then asks for your network ID:

Network ID:

Step 5 Enter your PennNet network ID and, when prompted, your network password.

If your network ID and password are correct, PennNet displays the annex: prompt.

If you make a mistake entering your network ID or password, PennNet lets you try again. After three tries, however, PennNet logs you out and you must start over again with step 2.

Step 6 From the annex: prompt, enter `telnet hostname.wharton` where *hostname* is the system you want to connect to.

For example, to connect to Wharton's Unix system Equity enter `telnet equity.wharton`. To connect to Wharton's VMS system Wilma enter `telnet wilma.wharton`.

From the annex: prompt you can also connect to non-wharton systems, such as Penn's library (`telnet library`), PennInfo (`telnet penninfo`) or any Internet host system.

Once you're connected to the host system you may see a brief message identifying the system. You are now ready to log on (see "Logging On"). When you are finished, be sure to log off the system (see "Logging Off").

Connecting from Wharton Offices

Many offices within the School have connections to Wharton's network, either through an asynchronous connection to a terminal server or a direct Ethernet connection.

If you're connected asynchronously through a terminal server, you'll typically see one of the following prompts—"DIAL:", "annex:", or "Local>". See the steps in "From an Asynchronous Connection" to connect to a host system.

If you have a direct Ethernet connection, you won't see a terminal-server prompt, and can connect directly from your computer to the host system using a TCP/IP connection. See "From an Ethernet Connection Using TCP/IP," below for more information.

From an Ethernet Connection Using TCP/IP

If you have a direct Ethernet connection to Wharton's network you can connect to a host system at Wharton (or anywhere on the Internet) using a communications protocol known as "TCP/IP."

How you connect to a host system depends on the type of TCP/IP software you're using.

► In most cases you can simply enter `telnet hostname`, where *hostname* is the name of the host system.

Depending on your TCP/IP software, you may use the command `tnvt hostname` or `tn hostname`. If you're using Windows or a Macintosh, you may have version of TCP/IP that allows you to select a host name from a menu or by entering it into a dialog box.

Contact your departmental computing support person if you're not certain how your TCP/IP software works.

To connect to a Wharton host from within the Wharton School, you only need to specify the system's name. For example, to connect to system Futures enter **telnet futures**.

To connect from a remote Internet site, you need to give the full Internet name of the system (referred to as a "fully-qualified domain name"). For example, to connect to Futures from another Internet site, enter **telnet futures.wharton.upenn.edu**.

From an Asynchronous Connection

To connect to a Wharton host system from a Wharton office with an asynchronous connection, do the following:

Step 1 Start your communications software.

Refer to the manual for your communications software for more information.

Step 2 Press the Enter key several times to get to the connection prompt. If requested, type in your username and press the Enter key.

Depending on how your office is connected, the screen should show one of the following prompts:

DIAL: (Go to Step 3.)

or

annex: (Go to Step 4.)

or

Local> (Go to Step 5.)

Step 3 If the screen says DIAL: enter **telnet**.

The screen says:

RINGING

ANSWERED

► Press the Enter key a few times to display the Network ID: prompt.

PennNet responds with a connect message that begins Annex Command Line Interpreter, and then asks for your network ID:

Network ID:

► Enter your PennNet network ID and, when prompted, your network password.

If your network ID and password are correct, PennNet displays the annex: prompt; continue with the next step.

Step 4 If the screen says annex: enter `telnet hostname.wharton` where *hostname* is the system you want to connect to.

For example, to connect to Wharton's Unix system Equity, enter `telnet equity.wharton`. To connect to Wharton's academic VMS system, enter `telnet wilma.wharton`.

Step 5 If the screen says Local>, enter `c hostname` where *hostname* is the system you want to connect to.

From the Local> prompt, you can only connect to Wharton's VMS-based systems, such as host Wilma. To connect to Wilma, for example, enter `c wilma`.

After completing step 3, 4, or 5 you should be connected to the host system. Depending on the system, you may see a brief message identifying the system. You are now ready to log on.

Logging On

Once you connect to a system you'll typically see a brief connection message that identifies the system, and will then be asked to identify your account or username.

A Unix system displays the login prompt:

login:

On a VMS-based system you're asked for your username:

Username:

Step 1 Type in the user name you were assigned when you opened your account, and then press the Enter key.

On a Unix system, user names and passwords are case sensitive, and must be entered with the correct capitalization.

For most accounts, you need to enter a password:

Password:

Step 2 Type in your password and then press the Enter key.

The password does not appear on the screen when you type it.

If you make an error entering your username or your password, you'll see an error message such as "User authorization failure" or "Login incorrect." If this happens, press the Enter key if necessary to return to the Username: or Login: prompt, and repeat steps 1 and 2.

If you enter your username and password correctly, you will usually see a log on message.

On some systems, you'll be asked to select a terminal type:

TERM = (vt100)

Unless you're using a special type of terminal emulation, you can either enter `vt100` or simply press the **Enter** key.

Once you're logged on, you'll see the system prompt—typically a percent sign (%) for a Unix system or a dollar sign (\$) for a VMS system. On some systems, the system prompt may include the name of the system or the current working directory.

On some Unix systems you'll see an opening menu listing many commonly-used commands. From the Unix system prompt (%), you can enter `menu` to redisplay this list of commands.

On certain systems (like Wharton's VMS systems), you are required to change your password when you log on for the first time. On other systems you may want to select a new password (particularly if one was automatically assigned to you when the account was created). See the next section for more information on changing passwords.

When you are finished using the system, be sure to log off before you leave (see "Logging Off," below).

Changing Your Password

When you log on to Wharton's academic VMS systems for the first time, you must change the password you were assigned. Follow the instructions below beginning at Step 2.

Whatever system you're using, you may want to select a new password when you first use the system.

The exact steps for changing your password depend upon the system you're using. On most Wharton systems, you can change your password as follows:

Step 1 At the system prompt, enter the following:

- ▶ On a Unix system, enter `passwd` at the Unix system prompt (%)
- ▶ On a VMS system, enter `set password` at the VMS prompt (\$)

You'll then be prompted for your old password.

Step 2 When asked for your old password, type in your current password and then press the Enter key.

You'll then be asked for the new password you'd like to use.

Step 3 Type in a new password.

Your password should be at least 6 characters long. On Unix systems passwords are case sensitive—lower-case and upper-case letters are different. To avoid problems, it's a good idea to use only lower-case letters, numbers, and the underscore, period, or exclamation mark characters. Do *not* use the pound sign or the "at" sign in your password.

The password does not appear on the screen when you type it.

To make sure you typed your new password correctly, you are asked to enter it again.

Step 4 Type in your new password again.

If you typed the same password both times, the system returns to the system prompt. From now on, use this password to log on to this system.

If you make a mistake typing your password, you'll receive an error message, and the system does not change your password. If this happens, repeat steps 2 through 4.

Logging Off

When you are finished using any networked host system, make sure you log off the system when you're finished.

Step 1 At the system prompt, enter the following:

- ▶ On a Unix system, enter `logout` at the Unix system prompt (%)
- ▶ On a VMS system, enter `logoff` at the VMS prompt (\$)

Step 2 If you connected to the system asynchronously through a terminal server, log off the terminal server:

- ▶ If your screen says `annex:` type `hangup` to log off the server.
- ▶ If your screen says `Local>` type `logout` to log off the server.

Step 3 Exit your communications software.

How you do this depends on the software you're using.

- ▶ If you're in Wharton's DOS/Windows labs and you selected a host from a Program Manager icon, when you log off you will automatically exit the Telnet program and return to Windows.
- ▶ If you're in Wharton's DOS/Windows labs and you selected the Telnet icon and entered a host name in the "Open Session" dialog box, when you log off the host system, the "Open Session" dialog box reappears. To return to the Windows Program Manager, Click the "Cancel" button and then select Exit from the File menu.
- ▶ If you're using MS-Kermit, press `Alt x` to exit Kermit and return to DOS.
- ▶ If you're using the DOS version of ProComm, press `Alt x` and then type a `Y` when asked "EXIT TO DOS?"

If you're using a different communications package, consult your manual.

Step 4 If you are finished using a computer in Wharton's labs, exit Windows and turn off the computer.

Files and Directories: Wharton's Unix Systems

File and Directory Names

Unix file names are limited to 14 characters, and certain nonalphabetic characters cannot be used in the file name. To avoid problems, it's a good idea to use only letters, numbers, the underscore character, and the period when naming your files.

Unlike DOS, VMS, and most other operating systems, Unix is case sensitive—lower-case and upper-case letters are different. This means that `myfile`, `MyFile`, and `MYFILE` are three different file names and must be typed with the correct capitalization.

File names that begin with a period are "hidden" files and are not normally displayed. For example, most users have the file `.login` in their home directory, but this file is not displayed with the `ls` command. *Do not remove or modify hidden files unless you are familiar with their purpose.*

Specifying Files and Directories with Path Names

When you specify a file by its name only, it refers to a file in the current directory. To refer to a file in different directory, you need to include the *path name* for the file.

An *absolute path* to a file or directory lists all the directories from the top-level "root" directory down to the file or directory, with each directory name separated by a forward slash (/) character.

For example, `/users/welles/temp.txt` refers to the file `temp.txt` in the directory `welles` which is in the directory `users`.

You can also refer to a file or directory by using a *relative path*, which locates the file or directory relative to your current working directory or by using several abbreviations for a full path:

- . Your current directory.
- .. Up one level from your current directory.
- ~ Your home or login directory.
- / The top-level or "root" directory.

Moving Around the File System

You are always "in" a specific directory, which is used as the default directory for most commands.

pwd "Print working directory;" shows your current directory.
cd *dirname* "Change directory;" moves you to a new default or "working" directory.

Examples:

cd /usr/local/bin

Moves you to the directory /usr/local/bin (if it exists).

cd -

Returns you to your home directory.

cd ..

Moves you up one level to the "parent" of the current directory.

If you don't specify a directory when you use **cd**, you are placed in your home directory. (This is different from MS-DOS, where **cd** shows you your current directory. To see your current directory in Unix, use **pwd**.)

Displaying Files

To see the names of the files in a directory:

ls *filespec* Displays a list of the names of the files identified by *filespec*.

Examples:

ls

Displays the files in the current directory.

ls *.ps

Displays the files in the current directory with the extension .ps.

ls /usr/local/bin

Displays the files in the directory /usr/local/bin.

ls ~/w*

Displays the files in your home directory that begin with the letter w.

ls -a

Displays *all* the files in the current directory, including "hidden" files (files with names that begin with a period).

ls -l

Displays a *long* listing of the files in the current directory, which includes information on each file's access rights, owner, group, size, modification date, and name.

To see the contents of a file:

- cat filename** Displays the contents of a file.
- more filename** Displays the contents of a file one screen at a time. To display the next screen, press the space bar. To exit, press the q key.
- less filename** Displays the contents of a file one screen at a time, and allows you to scroll forward or backward in the file.
- To display the next screen, press the space bar or the f key. To scroll back to the previous screen, press the b key. To exit, press the q key.

Finding Files

To look for a file by name:

- find dir -name 'filename' -print**
Searches for the file *filename* in the directory *dir* or directories below that directory.

Examples:

- find ~ -name '*.for' -print**
Lists the names of all the files with the extension *.for* in your home directory, or any subdirectories beneath it.
- find / -name 'wharton.txt' -print**
Searches the entire disk (from the top-level "root" directory on down) for files named *wharton.txt*.

To look for a file based on its content:

- grep 'string' filename**
Lists all the files that contain the text *string*.

Examples:

- grep 'Wharton' *.txt**
Lists all the files with the extension **.txt* that contain the text "Wharton."

Moving and Deleting Files

- cp filename newfilename**
Copies a file.
- mv oldfile newfile**
Moves a file.
- rm filename** Deletes (removes) a file.
- Be careful when using wildcards to specify groups of files when using the **rm** command. Before using **rm** use the wildcard with **ls** to make sure you know which files will be deleted.

Creating Files and Directories

mkdir *dirname* Creates a new directory.

Examples:

mkdir homework

Creates a new directory *homework* underneath the current directory.

pico *filename* Creates or edits the file *filename* using the editor "pico."

To exit pico, press Ctrl x. If you've modified the file, type y to save your changes or n to exit without saving your changes.

emacs *filename* Creates or edits the file *filename* using the editor "emacs."

To exit emacs, press Ctrl x and then Ctrl c. If you've modified the file and you want to save your changes, type y. To exit without saving your changes, press n, and then type **yes** when asked Modified buffers exist; exit anyway?

For More Information

Most Unix systems have extensive on-line help, in what is referred to as "man pages." To get help on a particular command, enter **man *command***, where *command* is the name of the command.

For example, to find out about all the options available with the **ls** command, enter **man ls**. To find out more about the pico editor, enter **man pico**. To find out more about man itself, enter **man man**.

If you need help but are not sure of the command name, enter **man -k *keyword*** to look for help topics that include the word *keyword*.

The man program displays help information one screen at a time. Press the space bar to display the next screen. To exit the man program, press the q key.

elm: ELectronic Mail

What is electronic mail?

Electronic mail is a way to send messages over a computer network. E-mail messages travel across the network quickly. On campus, they usually arrive a few minutes after they are sent.

Who uses electronic mail?

Many students, faculty and staff and a growing number of organizations all over the world use e-mail. You can contact your colleagues and friends both inside and outside Penn using e-mail.

E-mail etiquette

Since e-mail messages are read on a computer screen, people often like to keep messages, paragraphs and sentences short. However, don't forget to be polite.

Use blank lines between paragraphs to improve readability.

Use asterisks for *emphasis* and capital letters for SHOUTING.

Show humor with a sideways smiley :-)

Always remember that the recipient of your e-mail is a human being. Please treat other people the way you prefer to be treated.

Help with mail

To get help, see the instructions in the packet you received from your school with your computer account.

Get Started
To login, use your "How to login" instructions. Then, at the % prompt, type:

elm

followed by

or or 

Use lower case when you type elm.

E-mail addresses

To find your e-mail address, type the command `myaddress` at the % prompt. Addresses have the form:

`steve@eniac.sas.upenn.edu`

user-ID, usually some form of name
"at"
1-3 word computer name, varies by school at Penn
Penn

U.S. educational network.
Other U.S. networks are gov, mil, org and com.

E-mail addresses contain a person's user-ID, "at," the computer name, the organization name, and the network name, with no spaces in between. To send mail to someone on the same computer, the user-ID (`steve`) is all you need for the address. Sending mail to a different Penn computer, use both the user-ID and the computer name (`steve@eniac.sas`) for the address.

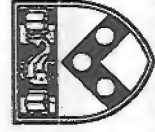
To find out another person's e-mail address, ask him or her what it is. Or, if you receive mail from the person, use the % command (see the "elm command reference" card). Penn is planning a database for e-mail addresses. Look for announcements about this in the *Penn Printout*.

Security

E-mail is like an electronic postcard: occasionally others might come in contact with messages you send. So don't send confidential information by e-mail.

Account policy

Don't share your Penn computer account. Use a secure password, and do not give it to others. See your school's policy for more information about uses of computers.



University of
Pennsylvania

SAS Computing
2/15/93

elm index and main menu

N = New Message
D = Message marked for deletion

Date	From	Subject
N 1 Sep 7	T.A. Leeds	Re: Calculus Question
2 Sep 6	Dr. Dean	New Student Reception
D 3 Sep 3	Elmo	Welcome to Penn

Current message is highlighted

Subject of mail message

Sender

Menu of commands

You can use any of the following commands by pressing the first character:
d) delete or u) undelete mail, m) all a message, x) reply or f) forward mail, q) quit.
To read a message, press <return>, j = move down, k = move up, ? = help

Command: █

When you start elm you will automatically see your mail in the index/main menu screen.

On the index screen, move the highlight 'up' among the messages with the up arrow or and 'down' with the down arrow or .

To read a highlighted message, press or or .
Press the to see the next page of a long message.
When finished reading, press to return to the index/menu screen.

To mail a message, press . You will be prompted like this:

Send the message to: elmo — Type the recipient's e-mail address here. In this case, it's only a user-ID, because elmo is at the same computer.

Subject of message: Hi, elmo! — Type a subject here, press .

Copies to: — Do you want to send copies?
If so, type e-mail addresses separated by spaces. If not, press .

Next, you will see the pico editor. Use pico to compose the text of your message (see "the pico editor" reference card).

Then when you're finished with pico, you'll see:

Please choose one of the following options by parenthesized letter:
a) dit message, edit headers, a) end it, or f) forget it.

Press to send, or to go back to pico and edit again, or to change the recipient's e-mail address and subject, or not to send the message at all.

To reply to a highlighted message, press .

Copy message? (y/n) — The highlighted letter is for: No, don't include a copy of the original message.

Press to accept this choice or press if you do want to include a copy.

Subject of message: — The subject of the original message will be put here, but you can change it by backspacing, then typing a new subject.

Next, you'll see the pico editor, the same as when mailing a new message. When you're finished with pico you'll be asked if you want to send, edit again, edit the headers, or forget it.

To delete the highlighted message, press .

Delete your messages when you are done with them.

To see more elm commands, press for help.

To quit from elm, press .

eln: working with folders

(How to organize your mail messages)

Elm shows your incoming mailbox first.

Mailbox is '/var/spool/mail/myuserid'

```
1 Jul 17 Jon Jones (15) RB: The dent in my car
N 2 Jul 17 Sarah Smith (12) Payroll question
```

Command: █

You can store a subset of your mail in other folders so you can view related messages together.

- To save the highlighted message to a folder, press **[s]**. You will be asked for the name of a folder.

save message to: =jones **[↵]** — Elm suggests a folder name which you can accept or change. To change, type **[=]** and a folder name (no spaces in between), then press **[↵]**.

The message you saved is deleted from the incoming mailbox.

Saved message now marked deleted. Highlight bar moves down.

```
D 1 Jul 17 Jon Jones (15) RB: The dent in my car
N 2 Jul 17 Sarah Smith (12) Payroll question
```

Command: █ Message 1 saved to folder /users/myuserid/Mail/jones.

Expanded form of =jones.

- Folder names can include letters, numbers, periods, underscores, or hyphens. Examples:

=payoh151 =ToDo Note that capitals count.
=pay.w-s =todo

- To change to another mail folder, press **[c]** at the Command: prompt. You will be asked for the name of a folder.

Change to which folder: =jones **[↵]** — Type **[=]** and a folder name, then press **[↵]**.

That quits the incoming mailbox, deletes messages marked D, and puts you in the =jones folder. Now you can see its index, and can issue elm commands.

Name of the folder you are viewing.

Folder is '=jones' with 1 message [ELM 2.3 PL11]

```
1 Jul 17 Jon Jones (15) RB: The dent in my car
```

Command: █

Only one message is saved in folder =jones so far.

- To change back to the incoming mailbox, press **[c]** at the Command: prompt. You will be asked:

Change to which folder: ! **[↵]** — Press **[!]** then **[↵]** (No leading **[=]** is necessary.)

That quits the current folder, deletes messages marked D, and puts you back in the incoming mailbox.

- To get help and list your folders, press **[c]** at the Command: prompt. You will be asked:

Change to which folder:

Type:

[?] **[↵]** for help and to list your folders.
[!] **[↵]** to change to your incoming mailbox (where you receive new mail).
[>] **[↵]** to change to your =received folder

(where some people save messages they have read already).

to change to your =sent folder

(where some people save messages they have sent to others).

[=jones] **[↵]** to change to folder =jones.

[↵] to return to the Command: prompt in the same folder.

You can type short aliases instead of long e-mail addresses.

Aliases work for groups of addresses, too.

- To begin working with aliases, press **a** at the elm index/main menu Command: prompt. Then you will see the Alias Commands menu.

You can use any of the following commands by pressing the first character:
a) alias current message, n) ew alias, d) elete or u) ndelete an alias,
m) all to alias, or r) eturn to main menu. To view an alias, press <return>.
j = move down, k = move up, ? = help

Alias:

Alias: prompt.

- Alias names can be any combination of letters, numbers, periods, underscores, or hyphens (no spaces).

- To make a new alias, press **n** at the Alias: prompt.

You will be asked:

Enter alias name: rob

Type an alias name that will be easy for you to remember, then press **Enter** or **Return** or **↵**

Enter last name for rob: Patel

Type the last name, **↵**, the first name, **↵**, then a comment if you want to, then **↵**

Enter first name for rob: Robby

Enter optional comment for rob: from Psych class

Enter address for rob: xpatel@sas.upenn.edu

Alias:

Robby Patel (rob) = xpatel@sas.upenn.edu

Accept new alias? (y/n)

Confirm the information. Press **n** to cancel or **↵** to accept.

Press **r** to return to the elm main menu. The new alias will be added to your personal database (alias list).

- Use the alias in place of the address at the Send the message to: prompt.

- To create a group alias, press **n** at the Alias: prompt. You will see:

Enter alias name: psychgroup — Type an alias name that will be easy for you to remember, then press **↵**

Enter last name for psychgroup: Psych 151 study group — Type a description for the group alias, then press **↵**

Enter first name for psychgroup: Press **↵**

Enter optional comment for psychgroup: Press **↵**

Enter address for psychgroup: rob, pavlov@doglab.com, freud@dolphin.upenn.edu — Type a list (using e-mail addresses or aliases) separated by commas, then press **↵**

There is a limit of 250 characters in any alias.

Alias:

psych 151 study group (psychgroup) = rob, pavlov@doglab.com, freud@psych.utex.edu

Accept new alias? (y/n)

Confirm the information. Press **n** to cancel or **↵** to accept.

- Other Alias commands. At the Alias: prompt, press:

? To get HELP.

d delete an alias.

a To make an alias for the sender of the current message.

f Show full list of addresses in the highlighted alias.

r return to the elm index/menu.

elm command reference

U of PA, SAS Computing 2/1/83

Where

Action

Command

Where

Action

Command

Working with mail:

SPACEBAR	Read current (highlighted) message	Index	
SPACEBAR	Page through message, at end display next message	Message	
←	Read current (highlighted) message	I	I, M
→	Scroll one line at a time, at end redisplay current message	M	I, M
n	Display current message	I	I, M
n	At end of message go to next one not marked deleted	M	I, M
m	Mail (write and send) an electronic mail message	I, M	I, M
r	Reply to the author of current message	I, M	I, M
g	Group Reply to the author and all recipients of the message	I, M	I, M
f	Forward current message with changes or additions	I, M	I, M
b	Forward message without changes (bounce)	I, M	I, M
d	Delete current message	I, M	I, M
u	Undelete current message	I, M	I, M
h	Display current message, showing full mail system header	I	I, M, T
h	Redisplay current message with full mail system header	M	

Additional features:

?	Help for elm commands	I, M
a	Access the commands to work with aliases	I
o	Change elm options	I
p	Print the current message (varies by school and computer location)	I, M, T

Exiting Elm:

q	Quit, deleting any messages that were marked for deletion	I
x	Exit, don't record as read, don't delete, ask permission if folder was changed	I, M
SHIFT-x	Quick exit, don't record as read, don't delete, don't ask permission	I

Working with folders:

a	Save current message to a folder	I, M, T
c =smith	Change to a folder named "smith"	I
c i	Change to the incoming mailbox	I
c ?	List your mail folders	I

Moving around an index:

↓ or j	Down to next message	I, M
↑ or k	Up to previous message	I, M
SHIFT-j	Down to next message, even if marked deleted	I, M
SHIFT-k	Up to previous message, even if marked deleted	I, M
↑ or +	Display next page of index	I, M
↓ or -	Display previous page of index	I, M
2 ←	Go to message number 2	I
=	Go to first message	I
*	Go to last message	I

Keystrokes

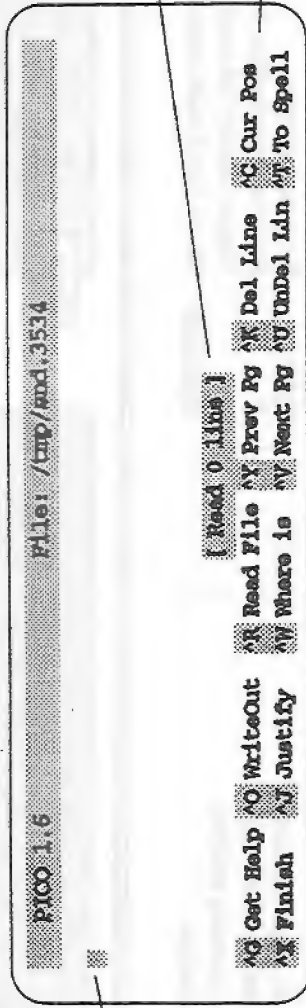
n means to press the **n** key.
ENTER means to press the **Enter** or **Return** or **↵** key.
CTRL-d means to hold down **CTRL** or **Control** and press **d**.
o i means to press **o** then **SHIFT** and **i** then **↵**.

Where

I ... commands used while viewing the index
M ... commands used while viewing a message
T ... commands used on tagged messages

the pico editor

Cursor position. Move it around using the arrow keys. When you start typing, the letters appear at the cursor position.



The symbol **^X** means to hold down the **[control]** or **[CTRL]** key then tap the **[x]** key. It works just as the **[shift]** key works: to get a capital **A**, hold down the **[shift]** key then tap the **[a]** key.

Now you can start typing the mail message. Text will appear at the cursor location.

^G Get Help Gives a list of all available commands.

^X Finish Saves the message in a temporary file, and puts you back into **elm**.

^O WriteOut Saves the message into a file. You will be prompted to type in a file name, and that file will be placed in your directory. File names cannot have spaces in them.

^J Justify Rearranges paragraph to fit screen width. Paragraphs are recognized in **pico** using blank lines or indentation.

^R Read File Reads in a file from your directory.

^W Where is Asks you for a word to search for in the message, and moves the cursor to the word. Or, you can search for a number or a string of letters.

^Y Prev Pg Moves to previous or next page (screen-full)
^V Next Pg of the message you are editing.

^K Del Line Deletes the entire line where the cursor is.

^U UnDel Lin Puts back the line where the cursor is.

You can use these two keys to move an entire paragraph or to duplicate a paragraph. First, place the cursor at the top line of the paragraph. Next, repeatedly type **^K** until the entire paragraph is deleted. Then, move the cursor to the paragraph's new location. Last, type **^U** to undelete the paragraph. If you type **^U** again, another copy of the paragraph will appear.

The next time you use a **^K**, **pico** will discard the paragraph, and start with newly deleted material.

^C Cur Pos Reports the position of the cursor.

^T To Spell Runs a spelling checker program on the message.

Other commands can be found if you type **^G**. Here is one of them:

^L Refresh Redraws the screen. If a message is sent to your terminal, such as a system shutdown notice, it may appear in the middle of your mail message. **^L** will remove it.

tin: NetNews Reader

What is NetNews?

NetNews is a way to post messages (articles) to be read by many people through Internet, similar to bulletin board systems. It is divided into discussion topics, called newsgroups.

Who uses NetNews?

Many students, faculty and staff and a growing number of people throughout the world use NetNews.

Moderated newsgroups

Most newsgroups allow anyone to post messages, but some groups have a moderator who selects appropriate postings.

NetNews etiquette

- ◆ Don't forget that the person on the other side is human.
- ◆ Be careful what you say about others.
- ◆ Be brief.
- ◆ Read a group before posting, to make sure your posting is appropriate.
- ◆ Use descriptive subject titles.
- ◆ Be careful with humor and sarcasm.
- ◆ Read all follow-ups before posting so you don't repeat what has already been said.
- ◆ Be careful about copyrights and licenses.
- ◆ Your postings reflect upon you; be proud of them.

Help with tin

To get help, see the instructions in the packet you received from your school with your computer account.

Get Started
To login, use your "How to login" instructions. Then, at the % prompt, type:

tin

followed by

(enter) or (return) or (↵)

Use lower case when you type tin.

Newsgroup names

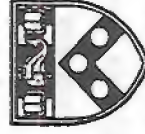
To find interesting newsgroups, start by looking at newsgroup names:

`upenn.sas.deans-forum` — discussion group with the SAS dean
`Penn School of Arts & Sciences`
`clari.biz.economy.world` — news from UPI, so it's copyrighted
`upenn.cis.cse120` — department course number
`upenn.forsale` — This is a typical course newsgroup.

Organization of newsgroups

The top level of the newsgroup hierarchy has:

- ◆ `upenn` Local to University of Pennsylvania
- ◆ `clari` United Press International (do not redistribute—copyrighted)
- ◆ `pa` Local to state of Pennsylvania
- ◆ `news` NetNews topics and software
- ◆ `rec` Recreational activities, hobbies, arts
- ◆ `comp` Computer science and related topics
- ◆ `sci` Scientific research and applications
- ◆ `talk` Debate on controversial topics
- ◆ `alt` Alternative newsgroups
- ◆ `soc` Social issues



University of
Pennsylvania

This group is for selling your personal belongings, but not for business postings, since Penn facilities are not commercial. "For sale" postings should only go in this group, not in other upenn groups.

`misc.jobs.offered`

Only for job ads, with heavy emphasis on jobs in computing of all sorts.

`upenn.talk`

Discussion of issues at Penn.

`rec.sport.soccer`

`alt.folklore.urban`

`sci.math`

`alt.chinese.text`

There are thousands of newsgroups available on NetNews. These are some of the most popular groups subscribed to by people at Penn.

Selecting newsgroups

The first menu shows newsgroups to which you are already subscribed.

unsubscribed
group Number of
unread articles Description
(if any)

	Group selection	
1	olari.feature.dave.barry	Columns of humourist Dave
2	upenn.talk	
3	misc.jobs.offered	Announcements of positions
4	sci.med.nursing	Nursing questions and answers
5	upenn.seas.general	
6	rec.humor.funny	Jokes that are funny (in
7	rec.arts.sf.tv	Discussing general television

Use these commands by pressing the first character: a)unsubscribe, u)unsubscribe, g)catchup, h)help, m)move, q)quit, TAB=next unread, j=down, k=up, /search pattern, y=yank in unsubscribed groups

↑ Move the highlight (selected group) down.

↑ Move the highlight up.

→ Go in to the highlighted newsgroup. Note: this command moves into the "Article Level" and you will see the commands at the right.

⇐ Quit out of tin.

c catch up in the highlighted newsgroup, marking all articles as read.

g go to a particular newsgroup whose name you know already.

y Display (yank in) all of the unsubscribed newsgroups, marked with "u" for unsubscribed. Now you can subscribe to new groups.

s subscribe to highlighted newsgroup (see also "y" command).

u unsubscribe to a particular newsgroup.

/ Search displayed newsgroups for a pattern, such as /humor

h help, and more information.

Postings and followups

A "posting" is a new topic. "Followups" are responses to articles already posted. Each posting is grouped together with its followups into threads.

When you use the command for postings or followups, you will be placed in the pico editor. See the pico reference card for instructions. Inside pico, you will see "headers," including the name of the newsgroup, the subject, and a blank line. Start composing your posting after the blank line. The blank line is required, so don't delete it.

Articles in a newsgroup

Articles and their responses are grouped into "threads," shown in this menu.

+ means
unread number of responses
in the thread subject person who posted the first
article in the thread

	upenn.talk	
1	What time is the rally today?	Patrik G. Matthew
2	5 Privacy	Meng Weng Wong
3	2 Community Outreach	Jeremy Anthony Ch...
4	Parking Permits for Hi-Rises?	Peter Harvit
5	Printing costs	hoop
6	2 Community Service-Soup Kitchen	Jennie Rosenbaum

1) list thread, TAB=next unread, /search pattern, a)author search, c)catchup, j=down, k=up, k=mark read, m)mail, q)quit, h)help, r=show/don't show unread, s)save, t)eg, w=post

↑ Move the highlight (selected thread) down.

↑ Move the highlight up.

→ Go in to read the highlighted thread. Note: new commands (below) become available while reading articles.

⇐ Go out of this newsgroup, back to the first menu.

c catch up, marking all articles in this newsgroup as read.

l list articles in the thread, showing all names of people who posted.

s save. You have a choice of saving the current article, the whole thread, or a group of articles. You will be asked for a file name.

w When you are asked for a "process," press Return for "none."

w The file will be saved in a directory called News.

w Post an article, beginning a new thread.

h help, and more information.

Commands while reading articles

↑ Page down through the articles in the newsgroup.

↑ Page back up through this article.

→ Go in to the next unread article in the thread.

⇐ Go out, back to "Articles in a newsgroup" menu, shown above.

m mail the article to someone.

r reply by mail to the author of the article.

f post a followup message, responding to the article.

NetNews Using "tin": Quick Reference

When you're using the `tin` news reader on Wharton's Unix systems, you're typically at one of three places:

- ▶ the newsgroup level,
- ▶ the article directory (also referred to as the "index page"), or
- ▶ within a specific news article.

Below are some of the commands commonly used at each of these levels. As with most Unix commands, case is significant. For step-by-step instructions on using `tin` to read NetNews, see the WCIT *TechBrief* "NetNews Using 'tin': Wharton's Unix Systems."

Entering and Exiting NetNews

`tin` Starts NetNews using the "tin" news reader.

Q Leaves the news reader and returns you to the Unix system prompt (%) or the Main Menu.

At the Newsgroup Level

↑ ↓ Moves you up or down one line through the list of newsgroups to select the current newsgroup. (If the down and up arrow keys don't work on your system, you can also use the `j` and `k` keys.)

j Moves you down one line (like the down arrow key on most systems).

k Moves you up one line (like the up arrow key on most systems).

Pressing the space bar Moves you down one screen through the list of newsgroups.

Ctrl f Moves you down (forward) one screen through the list of newsgroups (similar to the space bar).

Ctrl b Moves you up (back) one screen through the list of newsgroups.

h Displays help screens of the newsgroup level commands.

Pressing Enter (↵) Takes you to the article directory for the current newsgroup.

s Subscribes to the current newsgroup. The newsgroup will be displayed when you use `y` to display only subscribed groups.

u Unsubscribes to the current newsgroup. The newsgroup will be displayed when you use `y` to display all groups.

- y Switches the newsgroup listing back and forth between showing all the newsgroups and only those to which you have subscribed.

At the Article Directory Level

- ↑ ↓ Moves you up or down one line at a time through the article directory. (If the down and up arrow keys don't work on your system, you can also use the j and k keys.)

- j Moves you down one line (like the down arrow key on most systems).

- k Moves you up one line (like the up arrow key on most systems).

Pressing the space bar Moves you down one screen through the list of article titles.

- Ctrl f Moves you down (forward) one screen through the list of articles (similar to the space bar).

- Ctrl b Moves you up (back) one screen through the list of articles.

- l Lists the titles of the individuals articles within the current thread.

- h Displays help screens of the commands available at the article directory (or "index page").

Pressing Enter (↵) Displays the current article.

- q Moves you up to the newsgroup level.

From Within an Article

Pressing Enter (↵) Displays the next screen of the current article or, if you're at the end of the article, displays the next article.

- ↑ ↓ Moves you up or down one screen at a time through the current article. At the end of the article, takes you to the next article in the thread or—if there are no other articles—to the next thread. (If the down and up arrow keys don't work on your system, you can also use the Ctrl f and Ctrl b keys.)

- Ctrl f Moves you down (forward) one screen through current article (similar to the space bar and the down arrow key).

At the end of the article, takes you to the next article in the thread or—if there are no other articles—to the next thread.

- Ctrl b Moves you up (back) one screen through the current article (similar to the up arrow key).

Pressing the space bar Moves you down one screen through the current article. At the end of the article, takes you to the next article in the thread or—if there are no other articles—to the next thread.

- Pressing Enter (↵) Moves you down one screen through the current article (like the space bar). At the end of the article, takes you to the next article in the thread or—if there are no other articles—to the next thread.
- h Displays help screens of the commands available within an article.
 - q Moves you up to the article directory level.
 - r Sends electronic mail to the author of the current article. Includes a copy of the current article.
 - R Sends electronic mail to the author of the current article.
 - m Mails a copy of the current article to another user.
 - f Posts an article to the newsgroup in response to the current article. Includes a copy of the current article in the new article.
 - F Posts an article to the newsgroup in response to the current article.
 - w Posts a news article on a new topic. (To post a reply to an existing article, use f or F.)
 - s Makes a copy of the current article in a Unix file.
 - q Moves you up to the article directory level.

Getting Help for Computing Problems

Walk-in and Telephone "Hotline" Support

WCIT Consulting

212 VH
 898-8600

Wharton Computing and Information Technology (WCIT) provides computer consultants to assist students, faculty, and staff in using Wharton's computer systems, software, and services.

- ▶ *Wharton computer consultants, walk-in help:* 212 Vance Hall.

Standard hours: 9 AM to 12 noon and 1 PM to 5 PM,
 Monday through Thursday; 1 PM to 5 PM Friday.

Summer hours: 1 PM to 5 PM Monday through Friday.

- ▶ *Wharton consulting "hotline":* 898-8600.

Hours: The same as walk-in consulting hours (see above).

CRC

38th & Locust Walk
 898-9085

The University's Computing Resource Center (CRC) provides consulting for Macintosh and DOS/Windows systems, file-transfer services, and virus protection services.

- ▶ *CRC, walk-in help:* Locust Walk across from the Bookstore.

Standard hours: 9 AM to 4:30 PM, weekdays.

- ▶ *CRC telephone support:* 898-9085.

Standard hours: 9 AM to 4:30 PM, weekdays.

Electronic and On-line Help

Unix Help

Most Wharton Unix systems provide extensive online help through "man" pages:

- ▶ *Unix "man" Pages:* Type `man command` where `command` is the name of a Unix program or command for more detailed information on a particular command.

Type `man -k keyword` to search for information on `keyword`.

For more information on using the `man` command enter `man man`.

VMS Help

If you use Wharton's Academic VMS system (host system "Wilma"), you can get online help for many VMS procedures.

- ▶ *VMS Help:* Type `help` at the VMS system prompt (\$).

Many VMS applications have additional online help available from within the application. For example, for help using the VMS Mail utility, enter `help` from the Mail prompt: `MAIL> help`

Consultant E-Mail

You can also use electronic mail (e-mail) to send questions to the computer consultants at Wharton or the CRC.

- ▶ *Mail to Wharton's consultants:* send to username **consultant**.
- ▶ *Mail to CRC consultants:* send to username **crc@a1.relay.upenn.edu**.

You should receive an answer by e-mail within a day to two.

CRC Tutorials

898-9085

The CRC maintains a library of tutorial software that you can use to learn at your own pace. Most tutorials are at the introductory level.

- ▶ *CRC Tutorials:* 898-9085.

Computing Documentation**WCIT TechBriefs**

WCIT *TechBriefs* are short "how-to" documents that provide step-by-step instructions on a single computing topic. Additional *TechBriefs* contain information on computing services and policies.

- ▶ *WCIT TechBriefs:* 212 Vance Hall.

WCIT User Guides

WCIT *User Guides* contain information on selected software and systems used at Wharton, including Wharton's data resources such as Compustat, CRSP, and Citibase. WCIT *User Guides* are available for reference in Wharton's computer consulting office or can be purchased from Wharton Reprographics.

- ▶ *WCIT User Guides (Reference):*
Wharton computer consulting, 212 VH (898-8600).
- ▶ *WCIT User Guides (Purchase):*
Wharton Reprographics, 400 SH-DH (898-1251).

Vendor Documentation

The vendor documentation for software installed on Wharton's computer systems is available for reference in Wharton's computer consulting office.

- ▶ *Wharton computer consulting:* 212 VH (898-8600)

The CRC contains a wide selection of vendor documentation, primarily for microcomputer software (both MS-DOS and Macintosh).

- ▶ *Computing Resource Center:* Locust Walk across from the Bookstore (898-9095).

Third Party Books

In addition to the "official" manuals that ship with the software, third party documentation is available for most popular computer software.

Two bookstores with a wide selection of computer books are:

► *University Bookstore*: 3729 Locust Walk (898-7595).

► *Borders Bookstore*: 1727 Walnut Street (568-7400).

News Announcements

The following periodicals provide current information on computing events and services at the University:

► *Penn Printout*

Information about computing at the University appears regularly in *Penn Printout*. The September issue usually contains an overview of University resources for help with computing. Back issues of *Penn Printout* are available from the CRC.

► *The Wharton Journal*

Wharton Computing occasionally publishes news announcements and general information through articles in the *Wharton Journal*.

Training ClassesWCIT Short Courses

400 SH-DH
898-2667

Wharton Computing and Information Technology (WCIT) provides a series of computing "short courses" that offer hands-on computer training.

► *WCIT Short Courses (Registration)*: 400 SH-DH (898-2667)

WCIT computing short courses are free to Wharton students, staff, and faculty. Each course is \$25 for other members of the University community. A \$5 cash deposit may be required of Wharton affiliates who have previously failed to attend a course.

For course descriptions, see the WCIT *TechBrief*, "Computing Short Courses: Course Descriptions." For a list of classes, see the Short Course Schedule *TechBrief* for the current semester. Both documents are available from Wharton's computer consultants in 212 Vance Hall or at the WCIT Computing Services window in 400 SH-DH.

Lippincott

898-5924

At the beginning of each semester the Lippincott Library of the Wharton School provides one-hour training sessions on library research techniques, covering both print and online resources. In addition, daily one-hour training sessions introduce users to the electronic information services available at Lippincott, including both online services and CD-ROM systems.

► *Lippincott Training Classes*: 898-5924

CRC
898-9085

The Computing Resource Center offers both formal hands-on training courses and the more informal "Bits and Pieces" noontime seminars.

CRC hands-on courses are three-hour classes on various software topics for both the novice and advanced computer user.

► *CRC Training Classes:* 898-9085.

CRC's "Bits and Pieces" seminars are hour-long sessions held at noontime. Most sessions provide a presentation with time for discussion. Times and dates of the "Bits and Pieces" seminars are published in the *Penn Printout*. No registration is necessary.

You can profit from the MIS shortage

Ranks of computer grads are thinning as students lose interest

Investment houses go for broke hiring IS professionals

Demand for UNIX C programmers soars

There is going and will continue to be a severe shortage of college graduates coming into the information management field. Firms are trying to distinguish themselves in tough global markets by offering customers fast, easy access to timely information, so information management is and will remain a top, recession-proof priority. In fact, systems analysis is projected as the seventh fastest growing occupations in all white-collar and blue-collar sectors of the economy. Other hot job titles include programmer/analyst, business analyst, LAN manager, database manager, systems integrator, and MIS director.

The growing dependency of corporations on computers requires that today's managers understand the capabilities and limitations of computer systems and that they be actively involved in the design and implementation of those information systems. In response to that need the Operations and Information Management Department offers a rich set of courses in the area of management information systems including a concentration within the undergraduate program. In contrast to programs in computer science - which tend to focus on the development of basic computer technology - Information Management emphasizes the business use of technology. The primary objective is to improve the application of technology to business problems and to empower students to use computers as effective business tools.

OPIM 210 Management Information Systems

The course introduces the use and management of information systems. It counts as a general breadth requirement in the Wharton undergraduate program. This IS NOT a lab course like OPIM 101! No prior technical background is required. Classes focus on readings and cases.

OPIM 311 Business Computer Languages

The course introduces commercial programming practices and discusses the management of commercial software application development. Students gain practical experience using the programming language C.

OPIM 314 Computer Mediated Communication: Business, Technology, and Policy

Students gain both hands-on experience and knowledge of how companies are using the Internet to reduce costs, decrease cycle times, and support joint ventures. This course examines how companies manipulate, store, communicate and retrieve information to conduct business.

OPIM 315 Database Management Systems

This course introduces fundamental concepts and principals of data management and document retrieval. Students gain practical experience in the development and use of database and multimedia systems.

OPIM 410 Decision Support Systems

The course presents an overview of expert systems and decision support systems technologies, trends, and products. It also explores the development of expert systems and decision support systems and prepares students to use leading-edge tools in this area.

Only you can prevent an MIS shortage

Ranks of computer college grads are thinning as students lose interest

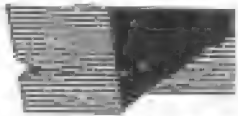
FREDERIC WITHINGTON



There is going to be a severe shortage of college graduates coming into the computer field at the entry level, and only you can alleviate it.

According to a Michigan State University study, the academic majors most in demand today are computer science graduates. Employers want to hire more of them than graduates of any other technical field.

Demand is likely to continue, or even intensify, because the central ranks of MIS departments are being depleted. Many of the first generation are reaching retirement age, and many of the younger specialists have been lost to information centers and user departments during recent decentralizations.



"While it is becoming a ubiquitous cliché, dispatching trained IBM mainframe environments is a number of key techniques," says Richard Winder, national director of the Information Systems Division at enrollment firm Robert Hall International, Inc.

Because of this shift, Winder says, there's a short supply of C programmers who can build corporate applications and communications systems.

"Good C programmers are hard to find," says Joe Havelka,

Demand for Unix C programmers soars

BY ALICE LAFARTE

Programmers who can write applications for Unix in the C language are getting the red carpet treatment. A strong demand for these professionals has been created for two reasons: Unix is becoming more popular in the commercial sector, and C is overwhelmingly the language of choice on Unix because of its portability across a wide variety of platforms.

Four years ago, Unix was confined mainly to scientific and engineering shops. But the increasing acceptance of IBM's AIX and San Microsystems, Inc.'s Sparc stations — coupled with the downsizing of distance-based applications onto minicomputers and local-area networks — has accelerated interest in Unix in the commercial sector.

Brokerage firm, manufacturer facilities and even banks are moving to take advantage of both the portability and cost savings inherent in a Unix-based, smaller systems environment.

"Unix is becoming a ubiquitous cliché, dispatching trained IBM mainframe environments is a number of key techniques," says Richard Winder, national director of the Information Systems Division at enrollment firm Robert Hall International, Inc.

Because of this shift, Winder says, there's a short supply of C programmers who can build corporate applications and communications systems.

"Good C programmers are hard to find," says Joe Havelka,

technical director of the advanced systems group at Pacific Bell in San Ramon, Calif. He is currently looking to hire C experts for internal expert systems development.

Other IS managers say the shortage of Unix C programmers is beginning to erode — thanks in large part to the fact that 10 and those 15 years

Investment houses go for broke hiring IS professionals

BY LIZZY GOTT

Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.



Investment houses go for broke hiring IS professionals. The reason is simple: they need them to manage their investments.

Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.



Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.

Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.

invested to C in classes and seminars, with an eye toward making them more versatile. "This will ultimately make programmers more upwardly mobile," Panella says.

In addition, companies with heavy number-crunching or statistical processing needs have taken to Unix in a big way. That's why some storage house LLI

programmers can have their pick of jobs at a time when workers with solid technical skills are lacking.

"We haven't experienced a slowdown in demand for Unix C programmers at all, even though openings for IBM mainframe programmers have dropped off considerably," says Bob Rich, associate director of the southern region of Source EPP, who acts as a lot of local firms building departmental applications around Unix minicomputers, particularly in

they are probably in evidence for, because of C's unique syntax.

As employers demand for Unix C programmers grows, the demand for Unix C programmers is growing. The reason is simple: they need them to manage their investments.

Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.

Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.

Investment houses and financial institutions are hiring IS professionals in large numbers. The reason is simple: they need them to manage their investments.

Outlook: Brokerage

Where the jobs are: Best opportunities are in the Northeast, especially metropolitan New York. Possible also in the Midwest and South.

As the market moves to the right, the demand for C programmers is growing. The reason is simple: they need them to manage their investments.

THE BEST JOBS IN AMERICA

What's better than being a doctor or an engineer? Becoming a computer analyst!

BY JERSEY GILBERT There are no more dreaded words in corporate America than these: "The system is down." Your boss is screaming, your clients are whining. What can you do? Call a computer systems analyst, that's what. Systems analysts are the indispensable people who install, customize and supervise computer operations at offices and factories across the nation. And now, with their services increasingly in demand, it's no surprise that they have the best job in America,

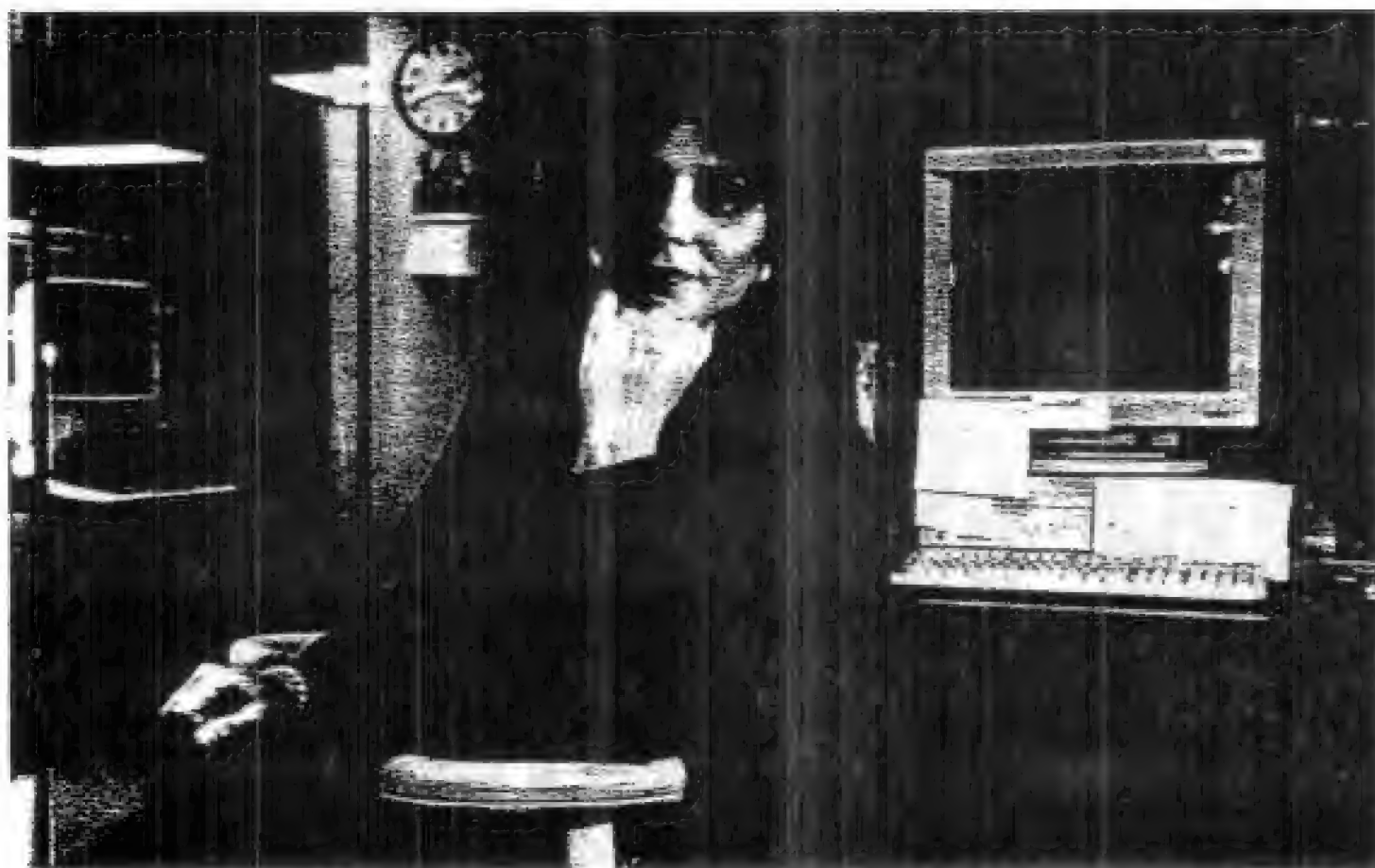
according to *MONEY*'s latest ranking of 100 jobs, chosen to represent a wide spectrum of pursuits. The Bureau of Labor Statistics (BLS) believes there will be 501,000 systems analysts jobs created between now and the year 2005, a gain of 110% from today's 455,000—and that 501,000-job forecast represents a 37% upward revision from just two years ago. That explosive anticipated growth helped propel systems analyst to the top of our chart from No. 31 in our previous jobs ranking, published in 1992. (Our complete listing of 100 jobs appears on page 72, and a guide to finding a new job when you lose yours is on page 74. Finally, on page 82, we present an exclusive poll that probes your attitudes toward work and enjoyment.)

Need more proof of our No. 1 choice? In her 10 years at the *New York Daily News*, Jean Leonardi, 36, pictured at right, has worked her way up from systems programmer to

system manager in charge of the software that prepares news copy for the printing press. Along the way, she has seen her salary rise above the industry median of \$42,700 a year. Leonardi adds that her work is satisfying as well. "In this job," she says, "people have a problem, you fix it, and boom, there's an immediate reward."

Among our other notable findings, doctors scored well despite all the talk of drastic health-care reforms. Their high prestige and salaries (median: \$148,000) lifted them to No. 2 on our list, up a notch from No. 3 in 1992. Two other health-care professions, however, rose sharply in the rankings thanks to the growing tendency to shift medical services away from high-priced M.D.s: physical therapist (No. 3, up from No. 50) and registered nurse (No. 28, up from No. 52).

Budget cutting at colleges and universities took a toll on some of the scientific careers that dominated our list



two years ago, including biologist (No. 16 this year, down from No. 1), geologist (No. 8, down from No. 2) and mathematician (No. 27, down from No. 4). Electrical and civil engineers, who are not dependent on universities for employment, more than held their own, moving up to the No. 4 and No. 3 spots from No. 13 and No. 9, respectively.

Even homemakers are not immune to economic trends. The Census Bureau reports recent declines in the percentage of couples who have children under 18 and those with only one wage-earning spouse. Those shifts led us to downgrade our estimate of the future demand for homemakers. Therefore, they fell to No. 61 from No. 51. Nonetheless, being a homemaker outranks several less-prestigious paying positions that involve some of the work that homemakers do for free, including cook (No. 72), waitress (No. 83) and telephone operator (No. 96).

Management consultants, on the other hand, jumped to No. 17 from No. 49 as job-cutting corporations increasingly turn to outside specialists to do work once performed by permanent staff; it's called outsourcing, in bureaucratese. And for all the talk of the Clinton Administration reining in lobbyists, their high salaries (median: \$91,300) and persistent influence vaulted them to No. 30 from No. 75.

Photograph by Michael Spano

SOFTWARE STAR

Systems analyst Jean Leonard spends her day making sure that news copy gets to the printing presses at New York's Daily News.

Trash collectors nosed out taxi drivers for last place. Along with earning the lowest prestige rating of any occupation on our list (through no fault of their

own), sanitation workers shoulder physical demands exceeded only by those of fire fighters and farmers. But fire fighters (No. 89), with median pay of \$32,200, are far better compensated than sanitation workers (\$18,800). Furthermore, the public, in an opinion survey, says being a farmer (No. 92) carries twice as much prestige as being a trash collector.

Our rankings are based on what you told us was most important about a job. In a recent survey of 250 MONEY readers, you rated opportunity for career advancement and job security at the top. Next came salary, having a clean and safe workplace, getting a chance to do something socially meaningful, prestige and avoiding stress. We gathered the data for our rankings from the BLS, the National Opinion Research Center, the *Jobs Rated Almanac* and more than 150 professional and industry organizations. \$

Reporter associates: Leslie Marable and Kelly Smith

JOB RANKINGS

Rank	1992 rank	Occupation	Median annual earnings ¹	11-year job growth	Short-term outlook	Job security rating	Prestige rating	Stress and strain rating ²	Where the jobs are
1	31	Computer systems analyst	142,700	110%	Excellent	Excellent	Good	Low	Silicon Valley, Washington, Boston
2	3	Physician	148,000	35	Average	Good	Excellent	High	New York, San Francisco, Philadelphia
3	50	Physical therapist	57,200	68	Excellent	Excellent	Good	Average	Denver, Boston, Seattle
4	13	Electrical engineer	69,100 ²	24	Good	Excellent	Good	Average	Silicon Valley, Dallas, Boston
5	8	Civil engineer	55,800 ²	24	Good	Excellent	Good	Average	Houston, San Francisco, Denver
6	7	Pharmacist	47,500	29	Good	Good	Good	Low	Columbus, Pittsburgh, Kansas City
7	29	Psychologist	53,000	48	Average	Average	Good	Average	Boston, San Francisco, New York
8	2	Geologist	50,800	22	Good	Excellent	Good	Average	Houston, Denver, New Orleans
9	15	High school teacher	32,500	37	Good	Excellent	Good	Average	Dallas, Houston, Atlanta
10	5	School principal	57,300 ³	23	Average	Good	Good	Average	Dallas, Houston, Atlanta
11	38	Paralegal	27,900	86	Average	Average	Average	Low	Washington, New York, Chicago
12	—	Hospital administrator	36,000 ⁴	36 ⁵	Fair	Average	Good	Low	Boston, Indianapolis, Philadelphia
13	—	Computer programmer	38,800	30	Good	Good	Good	Low	Washington, Silicon Valley, Dallas
14	12	Chemist	43,500 ³	21	Average	Average	Excellent	Low	Wilmington, Northern N.J., Raleigh/Durham
15	18	Dentist	93,000	5	Average	Good	Excellent	Average	New York, San Francisco, Seattle
16	1	Biologist	46,000	25	Fair	Fair	Excellent	Low	Raleigh/Durham, Washington, Boston
17	19	Management consultant	61,900	43 ⁶	Good	Good	Good	Average ⁷	Washington, Chicago, Minneapolis
18	46	Technical writer	37,400	23	Excellent	Excellent	Average	Low	Silicon Valley, Boston, Washington
19	19	Grade school teacher	31,000	21	Good	Excellent	Good	Average	Dallas, Houston, Atlanta
20	36	Construction superintendent	44,900	47	Good	Good	Good	High ⁸	Atlanta, Houston, Baltimore
21	11	Aerospace engineer	56,700 ³	14	Poor	Poor	Excellent	Low	Los Angeles, Seattle, Dallas
22	14	Bank officer	43,000	40 ⁹	Average	Average	Good	Low	New York, Los Angeles, Washington
23	56	Accountant	31,800	32	Good	Good	Good	Low	Washington, Dallas, New York
24	9	Sociologist	45,600	20 ⁹	Fair	Fair	Good	Low	Washington, Raleigh/Durham, Rochester, N.Y.
25	36	Estimator	41,200	25	Fair	Fair	Good	Low	Washington, New York, Chicago
26	34	Clergy member	26,000	30	Average	Average	Good	Average	Greenville, S.C., Birmingham, Charlotte, N.C.
27	4	Mathematician	42,700	8	Fair	Fair	Good	Low	Baltimore, Silicon Valley, Boston
28	52	Registered nurse	35,700	42	Good	Good	Good	High	Boston, Pittsburgh, Philadelphia
29	8	Urban planner	42,800	23	Fair	Fair	Average	Average	San Francisco, Minneapolis, Seattle
30	75	Librarian	31,300 ⁸	25 ⁹	Excellent	Good	Average	High ⁴	Washington, Sacramento, Albany
31	52	Dental hygienist	28,600	43	Good	Good	Average	Average	Minneapolis, Seattle, Detroit
32	—	Nutritionist	29,700	26	Excellent	Excellent	Average	Low	Boston, St. Louis, Philadelphia
33	37	Preschool teacher	18,400	54	Good	Average	Average	Average ⁸	Dallas, Houston, Atlanta
34	42	Medical lab technician	27,700	26	Average	Average	Good	Low	Philadelphia, Baltimore, Memphis
35	10	Veterinarian	46,900	33	Average	Good	Good	Average	Sacramento, Columbus, Kansas City
36	59	Forest ranger	29,400 ³	12	Average	Good	Average	Average	Portland, Ore., Seattle, Sacramento
37	25	Purchasing manager	40,200	14	Average	Good	Good	Low	Washington, Seattle, Denver
38	64	Social worker	25,600	40	Fair	Fair	Average	Average	New York, Boston, Philadelphia
39	69	Computer repairer	30,500	45	Good	Good	Average	Average ⁸	Atlanta, Silicon Valley, Dallas
40	72	Hotel manager	54,000	40 ⁹	Average	Average	Average ⁸	Average	Las Vegas, Orlando, Honolulu
41	39	Financial planner	55,100 ⁸	30 ⁹	Good	Good ⁸	Average	Average ⁸	New York, San Francisco, Chicago
42	22	Airline pilot	56,500 ³	35 ⁹	Good	Good	Excellent	High	Dallas, Atlanta, Denver
43	16	Lawyer	60,500	31	Fair	Fair	Excellent	High	Washington, New York, Chicago
44	58	Licensed practical nurse	22,500	40	Good	Good	Good	High	Cleveland, Tampa, San Antonio
45	84	Paramedic	28,100 ¹⁰	36	Good	Good	Good	Very high	Detroit, Los Angeles, Chicago
46	20	Architect	34,100	20	Average	Average	Excellent	High	Boston, Washington, Atlanta
47	87	Photographer	23,400	25	Average	Average	Average	Average	Los Angeles, New York, San Francisco
48	71	Flight attendant	26,300 ¹¹	51	Good	Good	Average	Average	Dallas, Denver, Atlanta
49	45	Personnel manager	31,100	25	Average	Average	Average	Average ⁸	New York, Washington, Denver
50	26	Graphic artist	25,800	23	Average	Average	Average	Average	Los Angeles, New York, San Francisco

Notes: ¹Unless otherwise stated, income is median 1992 earnings. ²Data exclude consulting income. ³For middle-school principals. ⁴Data include other health-care managers. ⁵For chemists with bachelor's or grad. ⁶For corporate lawyers. ⁷Includes conservation scientists. ⁸Upper estimate. ⁹Includes navigators. ¹⁰Average income. ¹¹1991 data. ¹²Excludes ratings of workplace environment, mental and physical stress. ¹³Includes instructors. ¹⁴Includes sal. and product designers. ¹⁵Includes writers. ¹⁶Includes directors. ¹⁷Figure is captain's salary and living allowances. ¹⁸For farm managers. ¹⁹For industrial managers. ²⁰For health where health-related jobs were rated "fair." ²¹For interior. ²²Rating for used car salespeople. ²³For health services. Sources: Bureau of Labor Statistics, Census Bureau, Bureau of Economic Analysis, Career Jitsu, Bureau of Economic Analysis.

Computer systems analyst tops our 1994 ranking of 100 widely held jobs evaluated on such factors as salary, prestige and security (see the story for details). This table shows the data we used to rank each job. In addition, the last column suggests where you might have the most luck finding a particular job by naming the metro areas with the highest concentration of people in each field. —J.G.

Rank	1992 rank	Occupation	Median annual earnings ¹	11-year job growth	Short-term outlook	Job security rating	Prestige rating	Stress and strain rating ¹²	Where the jobs are
51	34	Librarian	\$29,500	12%	Fair	Average	Average	Low	Washington, Boston, Raleigh/Durham
52	33	Fashion designer	29,600 ¹⁴	21 ¹⁴	Good	Average	Average	High	New York, Los Angeles
53	48	Bookkeeper	19,500	3	Average	Average	Average	Low	Denver, Minneapolis, Portland, Ore.
54	44	Advertising executive	44,300	36	Fair	Fair	Good	High	New York, Chicago, Atlanta
55	47	Travel agent	23,800	65	Average	Average	Average	Average	New York, Los Angeles, Chicago
56	27	Funeral director	36,500	18	Average	Excellent	Average	High	Scranton, Milwaukee, Pittsburgh
57	87	Stockbroker	40,700	33	Average	Average	Average	High	New York, Chicago, San Francisco
58	86	Fast-food manager	21,100	44	Average	Average	Average	High	Orlando, Los Angeles, Atlanta
59	—	Receptionist	16,400	34	Average	Average	Fair	Low	San Francisco, Minneapolis, Seattle
60	38	Air traffic controller	43,300	10	Fair	Average	Good	Very high	Long Island, Jacksonville, Memphis
61	31	Housekeeper	0	5 ⁸	Fair	Good	Average ²⁰	Average ⁸	Just about anywhere
62	83	Journalist	29,900 ²⁴	26	Average	Average	Good	High ⁵	Washington, New York, Boston
63	60	Property manager	26,600	35	Average	Average	Fair	Average	San Diego, Denver, Dallas
64	23	Musician	28,900	25	Fair ¹⁵	Fair	Average	High ⁴	New York, Los Angeles, Nashville
65	28	Police officer	32,900	13	Good	Good	Good	Very high	New York, Washington, Chicago
66	76	Machinist	26,600	-1	Average	Average	Average	Average	Milwaukee, Cleveland, Houston
67	81	Hair stylist	14,200	35	Average	Average	Fair	Average ⁴	Miami, Las Vegas, Phoenix
68	65	Actor	31,300 ¹⁴	54	Poor	Poor	Good	High	Los Angeles, New York, San Francisco
69	78	Carpenter	22,800	20	Average	Average	Average	High	Seattle, Miami, Baltimore
70	87	TV news reporter	21,400	25 ⁸	Poor	Poor	Good	Average ⁸	Los Angeles, New York, Washington
71	68	Plumber	27,000	8	Average	Average	Average	Average	Philadelphia, Houston, Baltimore
72	85	Restaurant cook	13,100	46	Fair	Fair	Fair	Average	Las Vegas, Honolulu, Orlando
73	17	Army officer	43,800 ¹¹	-20	Poor	Fair	Good ²¹	Average ⁸	Washington, Fayetteville, N.C.; Austin
74	61	Heavy equipment operator	22,000	11	Average	Average	Average	High	Charlotte, N.C.; Birmingham, Atlanta
75	81	Cashier	11,700	24	Average	Average	Fair ²	Average	Las Vegas, Orlando, New Orleans
76	77	Auto mechanic	21,900	23	Good	Good	Fair	High	Detroit, Houston, Los Angeles
77	43	Secretary	22,100	4	Fair	Fair	Average	Average	Washington, New York, Philadelphia
78	73	Public relations person	31,900	26	Fair	Fair	Average	High	Washington, Boston, New York
79	82	Waiter	23,600	15	Average	Average	Average	High	Houston, Detroit, Birmingham
80	—	Appliance salesperson	23,300	21	Average	Average	Fair ¹	Average	Dallas, Los Angeles, Atlanta
81	83	Surveyor	28,700	13	Fair	Average	Average	High	Seattle, Portland, Ore.; Houston
82	53	Teller	16,600	-4	Poor	Poor	Average	Average ⁸	New York, Los Angeles, Philadelphia
83	34	Waitress/waitress	12,000	36	Fair	Fair	Poor	Average	Las Vegas, Orlando, Detroit
84	41	Retail buyer	25,700	13	Fair	Fair	Average	High	Minneapolis, Atlanta, Chicago
85	70	Truck driver	23,100	27	Average	Average	Fair	High	Los Angeles, Houston, Atlanta
86	82	Insurance agent	28,400	15	Poor	Poor	Average	High	Chicago, Dallas, Hartford
87	79	Real estate agent	31,700	11	Average	Average	Average	Very high	Miami, Orlando, Seattle
88	55	Bank teller	15,200	-4	Fair	Fair	Average	Average	Chicago, New Orleans, Northern N.J.
89	62	Fire fighter	32,200	17	Fair	Average	Average	Very high	Boston, Providence, Oklahoma City
90	52	Apparel salesperson	13,600	21	Fair	Poor	Fair ⁸	Average	New York, Chicago, Los Angeles
91	69	Auto salesperson	25,800	21	Average	Average	Poor ²²	High	Oklahoma City, Dallas, Nashville
92	74	Farmer	20,600 ¹⁸	-21	Poor	Average	Average	Very high	Job is mostly in cities
93	96	Construction worker	19,700	17	Average	Average	Fair	Very high	Houston, Miami, Baltimore
94	90	Advertising salesperson	30,700	14 ⁵	Fair	Fair	Fair	High	New York, Chicago, Atlanta
95	89	Mail carrier	32,900	1	Fair	Average	Average	Very high	New York, Washington, St. Louis
96	85	Telephone operator	20,100	-28	Poor	Poor	Fair	Average	Dallas, Phoenix, St. Louis
97	—	Dancer	14,800	25	Poor	Poor	Average ²³	Very high	Las Vegas, New York, Los Angeles
98	97	Butcher ²⁵	18,400	-14	Average	Average	Fair	High	Chicago, Omaha, San Antonio
99	100	Taxi driver	16,200	18	Average	Average	Poor	Very high	New York, Washington, Las Vegas
100	98	Garbage collector	18,800	11	Average	Average	Poor	Very high	New York, Miami, Philadelphia

Sources: American Medical Association, National Society of Professional Engineers, Educational Research Service, American Chemical Society, Commission on Professionalism in Science, Association of Management Consulting Firms, Bank Administration Institute, American Mathematical Society, Foundation for Public Affairs, Bureau of Personnel Services, College for Financial Planning, Journal of Emergency Medical Services, Vassar Stars, U.S. Army Public Affairs, Census.

Management

EFFICIENCY

Business analysts trained in operations research can be a secret weapon in a CIO's quest for bottom-line results.

By Mitch Kertz



Ministry man. Perhaps you've seen one at a cocktail party, explaining that the business could dispense that crowd around the popular shrimp dip if she would just divide the dip into three bowls and place them around the room.

As he sketches the improved traffic pattern on the back of a paper napkin, you notice that his favorite word is "optimize"—a surefire sign he has been trained in the little-known fields of "operations research" or "management science."

These folks are driven to solve logistical problems, a trait that may not make them sparkle on the party circuit but may be exactly what today's information systems departments need to deliver more business value.

Experts say smart IS executives will learn to exploit the talents of these mathematical wizards in their quest to boost a company's bottom line.

"If IS departments had more participation from operations research analysts, they would be building much better, richer IS solutions," declares Ron J. Ponder, chief information officer at Sprint Corp. in Kansas City, Mo., and former CIO at Federal Express Corp.

Ponder and others say analysts trained in operations research or management science can turn ordinary information systems into money-saving decision-support systems and are ideally suited to be members of the business process re-engineering team.

"I've always had an operations research department reporting to me, and it's been invaluable. Now I'm building one at Sprint," Ponder says.



Sprint's Ron J. Ponder: Operations research analysts help build "richer IS solutions."

As someone who has a Ph.D. in operations research and who built the legendary package-tracking systems at Fedex, Ponder is a true believer in something that many IS professionals have never even heard of.

Mathematical reasoning

So what is operations research? It's the use of advanced analytical techniques (such as mathematical models) to improve or optimize the performance of an organization. Management science is virtually the same, only its papers have a higher ratio of text to equations. Together, they go by the acronym OR/MS.

In either case, OR/MS analysts just love to solve business problems—and the more complex the puzzle, the more they like it. A classic example is the crew-scheduling problem at United Airlines. How do you plan the itineraries of 4,000 pilots and 17,000 flight attendants when there is an astronomical number of com-

binations of planes, crews and routes?

The analysts at United came up with a client/server-based scheduling system, called Paragon, that seeks to minimize the amount of paid time that crews spend waiting for flights (CW, May 11, 1992).

The Fortran model even factors in constraints such as union rules and Federal Aviation Administration regulations. It is expected to save the airline at least \$1 billion a year.

Over the years, some of the best CIOs have had operations research backgrounds. For example, Joseph T. Brophy, the award-winning former CIO at The Travelers Corp., previously worked as an operations research mathematician on the Polaris submarine weapons system.

Operations research got its start in World War II, when the military had to make decisions about allocating scarce resources to various military operations. Since then, the analytic sciences have spread throughout business and government, from designing efficient drive-

Efficiency Chronicles, page 64

Efficiency Einsteins

CONTINUED FROM PAGE 39

thru window service for Burger King Corp. to ultra-sophisticated computerized stock trading.

Somewhere along the way, perhaps in the 1970s, the operations research and IS disciplines went on separate tracks.

"The IS profession has had less and less contact with the operations research folks... and IS lost a powerful intellectual driver," says Peter G. W. Keen, executive director of the International Center for Information Technology in Washington, D.C.

The split is ironic, considering that one of the first business applications for computers in the 1950s was to solve operations research problems for the petroleum industry. A technique called linear programming was used to figure out how to blend gasoline for the right flash point, the right viscosity and the right octane and in the cheapest possible way.

The 1990s may be an ideal time for the two disciplines to rebuild some bridges, Keen and other observers say. Today's OR/MS professionals are involved in a variety of ill-related fields, including inventory management, electronic data interchange, computer-integrated manufacturing, network management and practical applications of expert systems and neural networks.

Furthermore, each side needs something the other side has. OR/MS analysts need the corporate data to program their algorithms, and they need their algorithms plugged into strategic information systems.

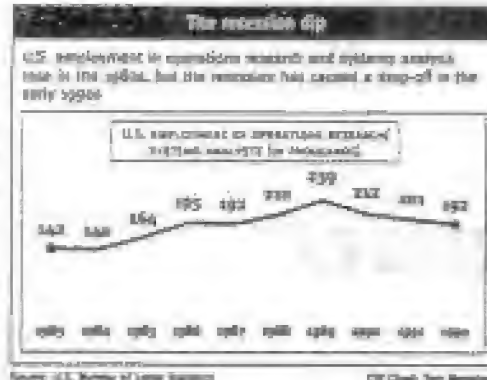
Meanwhile, CIOs need to build smart applications that enhance the bottom line and make them heroes with the chief executive officer.

Not people persons

However, Keen says, there are some barriers to collaboration. OR/MS professionals generally lack communication skills and sometimes focus on esoteric mathematics rather than real-world business problems.

"On the other hand, they are very, very bright people. If you can get them away from what I call 'rigor without relevance' and get them onto relevant projects, their rigor is very valuable," Keen says.

Perhaps the biggest barrier is an undercurrent of rivalry between some IS and OR/MS



groups as they compete for internal customers, budgets and glory. But failure to cooperate could be suicidal for both professions, experts say.

At a time when some operations research groups are facing budgets eroding from view altogether, and CIOs are getting fired left and right, it would behoove the two camps to



Joseph Murphy likes to blow his horn for OR/MS

cooperate on some CEO-pleasing "house rules," says consultant Donald B. Brown, president of Quality Technology Decisions, Inc. in New York.

"Operations research and management science have a lot to offer the CIO," says Brown.

who has a background in both management science and IS. "We can all be better,"

OR/MS analysts can develop a model of the way a business process works now and speculate how it could work more efficiently in the future, he says. Therefore, it makes sense to have an OR/MS analyst on the interdisciplinary team that tackles business process re-engineering projects.

In essence, OR/MS professionals add more value to the CIO infrastructure by building "tools that really help decision-makers analyze complex situations," says Andrew B. Whinston, director of

the Center for Information Systems Management at the University of Texas at Austin.

Thomas M. Cook, president of American Airlines Decision Technologies, Inc. in Fort Worth, Texas, puts it in even stronger terms. IS departments typically believe their job is done if they deliver accurate and timely information. But Cook says that adding operations research skills to the team can produce intelligent systems that actually recommend solutions to business problems.

One of the big success stories at Cook's operations research shop is a "yield management" system, which decides how much to overbook and how to set prices for each seat so that a plane is filled up and profits are maximized.

The yield management system, which deals with more than 250 decision variables, accounts for a whopping 14% of American Airlines' revenue. The airline's Sabre reservation system "got a lot of great press, but the value of things like yield management might even dwarf Sabre's benefits," Brown says.

Where to start

So how can the CIO start down the road toward collaboration with mathematicians?

Brown says that if the company already has a group of OR/MS professionals, the IS department can draw on their expertise as internal consultants. Otherwise, he says, the CIO can simply hire a few OR/MS wizards, throw a problem at them and see what happens.

The payback may come surprisingly fast. As one former OR/MS professional put it: "If I couldn't save my employer the equivalent of my own salary in the first month of the year, then I wouldn't feel like I was doing my job."



Resources

Operations Research Society of America, Baltimore, 4101 538-0346

The Institute of Management Sciences, Princeton, N.J., 609-683-2525

OR/MS Today, Atlanta, 404-433-0800. A bimonthly magazine (bimonthly) for operations research and management science professionals, with articles on practical applications.

Hollnagel & Co., Inc., One Greenway Drive, 1001 432-4813. Recruited in the field of IS, operations research and management science.

Analysis Consulting, Inc., New York, 212-687-9813. Performs operations research.

Home runs in management science

In the last decade, scores of operations research and management science projects have saved companies millions of dollars or have improved government services. Here are some of the "home runs," culled from the book *Excellence in Management Science*, Princeton (Princeton Hall, 1993).

1983: A computer-aided dispatching system at Chevron USA allowed each dispatcher to handle 400 loads per day, compared with the industry average of 150. Running on an IBM 3083, the system could solve a typical dispatching problem in less than a second. Benefit: Chevron cut transportation costs by 8%.

1982: General Dynamics Corp.'s Data Systems Division developed a model to help it make an IS capacity planning decision. The model showed that computer

throughput would be 25% higher if existing tape drives were replaced with drives that were 60% faster.

1982: The Arizona Department of Transportation developed a decision-support system for allocating maintenance funds to the toughest roads while staying within budget. The system has been adopted by Alaska, Colorado, Kansas, Poland and South Africa.

1984: The New York City Department of Sanitation used several mathematical techniques to improve the deployment of street cleaners, garbage trucks and inspectors, turning the department's embarrassingly poor performance into a national model. Benefit: Three-May or Ed Koch said the streets were much cleaner and

refuse collection productivity increased 17%.

1986: Wipac-Haweser Co. developed an integrative computer model that helps lumberjacks cut each log to maximize profits and minimize waste.

The model considers variables such as the log's length, diameter, curvature, taper and knots. Benefit: The company increased profits by \$100 million.

1987: General Motors Corp.'s Delco Electronics Division developed a PC-based planning system to identify the best way to ship 300 types of components to 60 assembly plants. Benefit: GM cut logistics costs by 25%, saving \$1.9 million a year. — *Mich Seta*



INFORMATION AGE / By WILLIAM M. BULKELEY

Computers Start to Lift U.S. Productivity

Computers are now making the U.S. work force more productive.

Of course, right? Well, economists found scant evidence of improved productivity from computers in the past decade. So, this is real man-bites-dog stuff.

Include Alan Greenspan, chairman of the Federal Reserve Board, among the believers. He told Congress last month that "a new synergy of hardware and software applications may finally be showing through in a significant increase in labor productivity."

Maybe it's synergy. Or maybe it's a new hard-headedness at banks, insurers and transportation companies that have fired people. Either way, more service sector companies are using computers to do the same amount of work with fewer people—the definition of productivity.

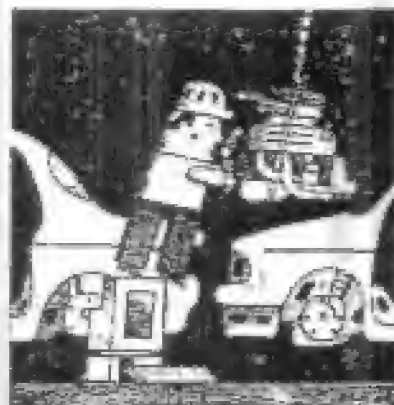
Fleet Financial Group Inc., a Providence, R.I., bank holding company, built a consumer service center in Utica, N.Y., to handle all customer inquiries from any of its seven bank subsidiaries in six states. Operating 24 hours a day, it gets 1.5 million calls a month—80% entirely handled by computer. Despite the longer service hours and wider range of inquiries handled, it now has 48% fewer customer service people than the separate banks did. "If you use technology to reengineer your business practices, there are significant gains," says Michael Zuehlke, executive vice president.

Working on the Railroad

Four years ago, Union Pacific Corp.'s Union Pacific Railroad unit closed 10 regional dispatch centers and consolidated control of the entire railroad in a \$50 million computerized center in Omaha, Neb. The action reduced the dispatch staff to 800 from 1,200. But "the real savings is in day-to-day operations," says Rip Hawley, vice president, transportation services. With the centralized system the railroad can reduce idle time for its 3,000 \$1.5-million locomotives by 1%, saving \$40 million a year. In the last nine years, the line has cut employment 30% to 27,832 while increasing tons hauled by 42%—a doubling in tons hauled per worker that partly reflects computerization.

During the past decade, overall U.S. productivity grew only 0.9% annually. Manufacturing concerns increased productivity 1.9% yearly, but in the service sector—where three-quarters of the work force is employed—growth was anemic.

Service sector companies "spent 1980 billion on information technology and got half a point of productivity growth a year," says Stephen Roach, a Morgan Stanley & Co. economist. "That's a horrible result for all the money spent." But Mr. Roach, who has railed against feckless computer in-



John Siegel

vestments for years, thinks he sees a change. "There has been a tremendous reversal of complacency. Managers are finally getting the long overdue payback from computing."

Gains for Economy

Any productivity payback is good for the economy in most ways. It holds down inflation. It provides a base for real increases in pay and profits. But companies' ability to do more work with fewer people may be one reason unemployment is falling so slowly. Ironically, computer programmers are among those frequently laid off, especially when companies merge and combine computer systems.

Analysts say there are a number of reasons that computers are finally improving productivity.

Some are technological. Mr. Zuehlke of Fleet Financial says that until the telephone companies installed high-capacity lines in the mid-1980s, it wasn't possible to transmit enough information from bank to bank to handle everything in one distant customer support center.

The spread of electronic data interchange, known as EDI, in which companies' computers exchange orders, invoices and payments electronically, instead of on paper, is reducing staffs. When Blue Cross & Blue Shield of Virginia's Health Communication Service unit developed a system most insurers could use to send payments directly to hospitals and doctors, University of Virginia Medical Center was able to cut the number of people keying vouchers into its computer system to seven from 14.

But most reasons are economic and cultural. Competitive pressure is making service companies eliminate costs—mostly people. Insurers that never reduced employment now frequently announce layoffs. Banks are consolidating, and executives of the surviving bank don't mind firing acquired employees and using technology to cut costs.

For years, service companies looked to computers mostly to improve service or improve internal processes. Frederick Sawyer III, senior vice president of Phoenix Home Life Mutual Insurance Co., Hartford, Conn., says, "During the '80s all the computer system work went into new [insurance] product development. Issues of administration and claims payment were getting short shrift."

Coping With Growth

But since Phoenix merged with Home Life last year, the company has emphasized better productivity. It redesigned the computer system used to answer customer questions, permitting a 37% cut in the total customer service staff to 194 from 316. Its mutual funds more than doubled new deposits in 1992 and 1991. But the customer service staff for mutual funds only increased to 65 from 60 because it installed computerized image systems to store customer correspondence, checks and instructions, eliminating time wasted looking for records in file cabinets.

Sometimes getting more productivity out of computers requires radical action. Salomon Brothers Inc. decided to move its support operations out of New York to Tampa, Fla., to remake its organization for handling trades, says Marc Sternfeld, managing director, U.S. operations. Despite adding new trading instruments, the Salomon Inc. unit is cutting support staffing to 500 from 670 and forcing traders in New York to enter their own trades on computers rather than slipping handwritten tickets into postmarked tubes for later processing. The move means that far fewer trades result in errors that have to be straightened out the next day.

Nobody thinks that most service companies have gone far on the road to improving productivity. Aetna Life & Casualty Co., Hartford, is just starting to eliminate processes that involve entering the same data manually in different computer systems. Pennell Hamilton, director of organizational effectiveness in small group health at Aetna, says field representatives—who had been using portable computers to print out reports on customers to be mailed to the head office—can now transmit the reports directly over telephone lines. "The application that gets missed a lot is the ability to enter data just once, at the source," he says.

"We're only at the beginning of the process," says John Skeritt, managing partner of Arthur Andersen & Co.'s consulting group, who advises financial services companies on using computers to improve operations. "Technologies like image processing, voice recognition, telephone banking, expert systems for doing loan evaluations, will continue to cause massive layoffs."

THE DIGITAL JUGGERNAUT



Never mind services—

the information economy is driving growth in the '90s. For most Americans, that translates into jobs and prosperity

BY MICHAEL J. MANDEL

In every era, there is a group of industries that sets the pace for the rest of the economy. A century ago, the railroads were America's growth engine. In the postwar decades, manufacturing was the key to U.S. prosperity. During the 1980s, the driving forces of expansion were booming service industries such as health care, legal services, and retailing. All told, during that decade, the service sector accounted for practically all of the growth in jobs and corporate profits. Economists began to speak of the U.S. shift from a manufacturing to a service economy.

Yet for all the vitality of services, many skeptics did not see how they could make the economy thrive over the long term. In fact, the shift seemed like a giant step backward, since service jobs paid lower wages on average than manufacturing and had significantly slower productivity growth. Moreover, services such as medical care and retailing were much harder to export than manufactured goods. The worry was that if the U.S. lost its manufacturing industries, it would have a difficult time selling

enough services abroad to pay for its imports of cars, consumer electronics, and other goods.

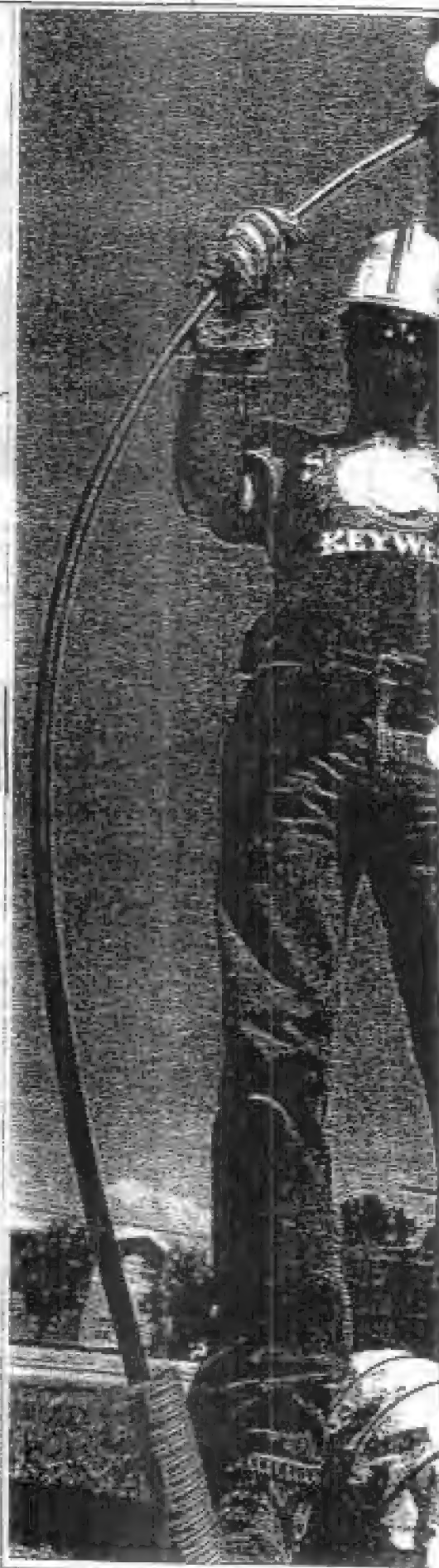
Fear not: Like adolescence, the service economy has turned out to be a temporary stage. Far more than most people realize, economic growth is now being driven not by services, but by the computer, software, and telecommunications industries. Indeed, according to the Commerce Dept., business and consumer spending on high-tech equipment accounts for some 38% of economic growth since 1990.

What's more, government statistics underplay the evolution of the information economy. Industries that depend on processing and moving information—such as financial services and entertainment—are prospering. And companies in every industry are using information technology to reengineer themselves and become more competitive. In short, "the role of information is transforming the nature of economy," says Kenneth J. Arrow, a Nobel prize-winning economist at Stanford University.

In this regard, at least, the U.S. is leading the way for the rest of the world. Europe is deregulating its telecommunications industry in order to create jobs and stimulate development. Japan is mounting an intense effort to narrow the considerable edge the U.S. has built over the decade in personal-computer and network use. Even developing countries such as China, Hungary, and Thailand are investing heavily in state-of-the-art communications systems in an effort to leapfrog their way to prosperity.

America remains way ahead, however. And it's the place where the conse-

REPAVING THE ROAD
NEW FIBER-OPTIC CABLES
ARE GIVING TELECOM
CAPACITY A BIG BOOST



quences of the new economy are first showing up. To a large degree, the news is turning out to be good. For one thing, unlike most services, information products such as software and entertainment can be easily exported. And whereas productivity in the service sector grew slowly, investment in information technology is boosting productivity across the economy.

Beyond that, the effect on work is less harmful than once feared. Far from becoming low-paid burger-flippers, the quintessential job of the service sector,

many Americans are turning into computer jocks. Economic studies show that their wages are on the rise as a result. For example, earnings for male computer programmers have risen by 12% since 1990, compared with 6% for all male workers. For female computer programmers, the pay gains have been even bigger: a 21% rise since 1990, vs. 13% for all female workers.

The drawback is that along with the winners, there will temporarily be lots of losers. Higher productivity has led to big layoffs at many companies, especially in the telecommunications industry (table, page 26). Elsewhere, meanwhile, advancing technology is favoring skilled workers over unskilled, increasing the inequality in wages.

For better or for worse, this transformation is occurring at an astonishing rate. Look at business investment. Measured in inflation-adjusted dollars, computers and other information technology now make up nearly half of all

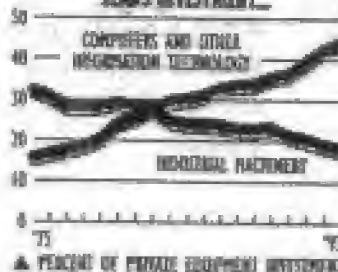
business spending on equipment—and that doesn't include the billions that companies spend on software and programmers each year. Meanwhile, business spending on industrial machinery, which traditionally has been the guts of manufacturing, has fallen as a share of equipment investment from 32% in 1975 to only 18% in 1993 (chart).

At the same time, information technology and services are helping to drive the continuing export boom. The aircraft industry is often held up as the shining star among U.S. exporters. Yet

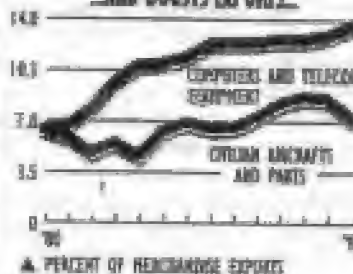
America's overseas sales of information-technology equipment in 1993 were \$62 billion, far more than the \$33 billion in overseas aircraft sales. The U.S. is also the world's largest exporter of software, a fact that doesn't show up in the government's numbers. In 1993, major U.S. software companies sold \$2.5 billion worth of personal computer programs in Western Europe, Asia, and Latin America, according to the Software Publishers Assn. Microsoft Corp. alone derives some 55% of its revenues from overseas sales.

The U.S. also is running a huge \$3 billion trade surplus in computer-related services, such as data processing and information databases. It's nearly as easy now to send information to Europe or Japan as to the next state or across the hall. For example, Mead Data Central Inc., the company that runs the Lexis and Nexis services, which contain legal news and general news respectively, also has databases on French and British

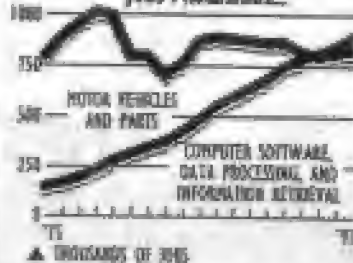
INFORMATION TECHNOLOGY LEADS INVESTMENT



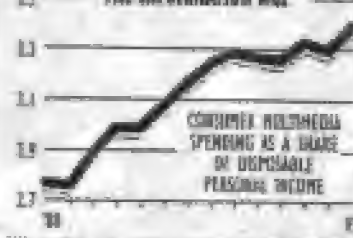
AND BOOSTS EXPORTS



WHILE INFORMATION JOBS PROLIFERATE



AND CONSUMERS JOIN THE INFORMATION AGE



SOURCE: COMPTON DATA, SHARE OF LABOR SURVEY, FEDERAL RESERVE & C.

law that lawyers in those countries use. The location of these databases: Dayton, Ohio.

Coming improvements in overseas communications will even make it possible to export such services as medical care. By this coming summer, doctors across sparsely populated South Dakota will be able to use a statewide telecommunications network to consult with specialists hundreds of miles away. The same expertise could be transmitted to Asia or Latin America just as easily. "The information economy can be a healthy economy because a lot of its services are exportable," says George Bennett, chairman of Symmetrix, a technology consulting firm.

Two other positive byproducts of the Information Age are greater efficiency and lower prices. During much of the 1980s, economists worried that they could not find any impact of computers on productivity. But more recent research shows that investments in computers are worthwhile. Economists Erik Brynjolfsson and Lorin Hitt of the Massachusetts Institute of Technology surveyed 400 large companies to gauge the effect of technology on output per employee. They found that the return on investment in information systems exceeded 30%. "And most of these benefits are being passed on to consumers in the form of lower prices," says Brynjolfsson.

In fact, the productivity surge of the last two years—when nonfarm output per worker rose by 4.9%, its biggest two-year jump since 1976—may reflect the efforts of U.S. companies to finally take full advantage of the huge sums they've spent purchasing information technology. "If I put technology in and nothing changes, and then later a business gets in a crunch and discovers that it can cut out all the middle management, what made it possible?" asks Raymond Perry, chief information officer at Avon Products Inc. "Well, probably the technology did. It's just that we weren't ready to take the people out until a later point in time."

Even the recent productivity numbers probably far

CASUALTIES OF THE INFORMATION ECONOMY

Computer and telecommunications companies that have announced job cuts within the last two years:

	WORKFORCE	
	PLANNED CUTS	PERCENT
IBM	35,000	14%
AT&T	26,000	9%
GTE	20,000	15%
NYNEX	17,000	22%
AMERITECH	11,000	15%
PACIFIC TELESIS	10,000	19%
BELLSOUTH	10,000	12%
U S WEST	9,000	14%
APPLE	2,500	16%
AST RESEARCH	1,000	16%
COMPAQ	1,000	10%
TOTAL	142,500	

SOURCE: BUSINESS WEEK

understate the critical role of information technology and services in driving growth. To put it simply: Government statistics track goods and jobs, not flows of information. That means the U.S. has a large and vibrant "ghost economy" that traditional economic indicators don't measure. Take the communications sector, which includes the telephone, broadcasting, and cable industries. According to government figures, communications is only 3.1% of the economy, up from 2.8% in 1984, at the time of the AT&T divestiture. Over the same period, minutes of telephone use—a key number tracked by the Federal Communications Commission—has grown only slightly faster than the overall economy.

Yet a closer look shows that the official statistics ignore many of the changes of the past decade. For one, a much greater percentage of the calls over the phone network are faxes and computer data going back and forth, rather than people talking. As much as 10% to 20% of the traffic across the AT&T long-distance network may be data, estimates Frank Ianna, the company's general manager for network services. That's up from 7% to 10% a few years ago. And because of time and language differences, about half of international calls are data, not voice.

These fax and computer messages pack a lot more data into a minute than they used to. Over the past few years, for example, the speed of a typical modem—which is used to transfer information between computers over phone lines—has quadrupled.

That means the amount of information being pumped through the system has gone through the roof. The point is this: If the output of the communications sector is measured in terms of data transferred instead of the number of minutes it's in use, it would show far more dramatic growth than the published numbers indicate.

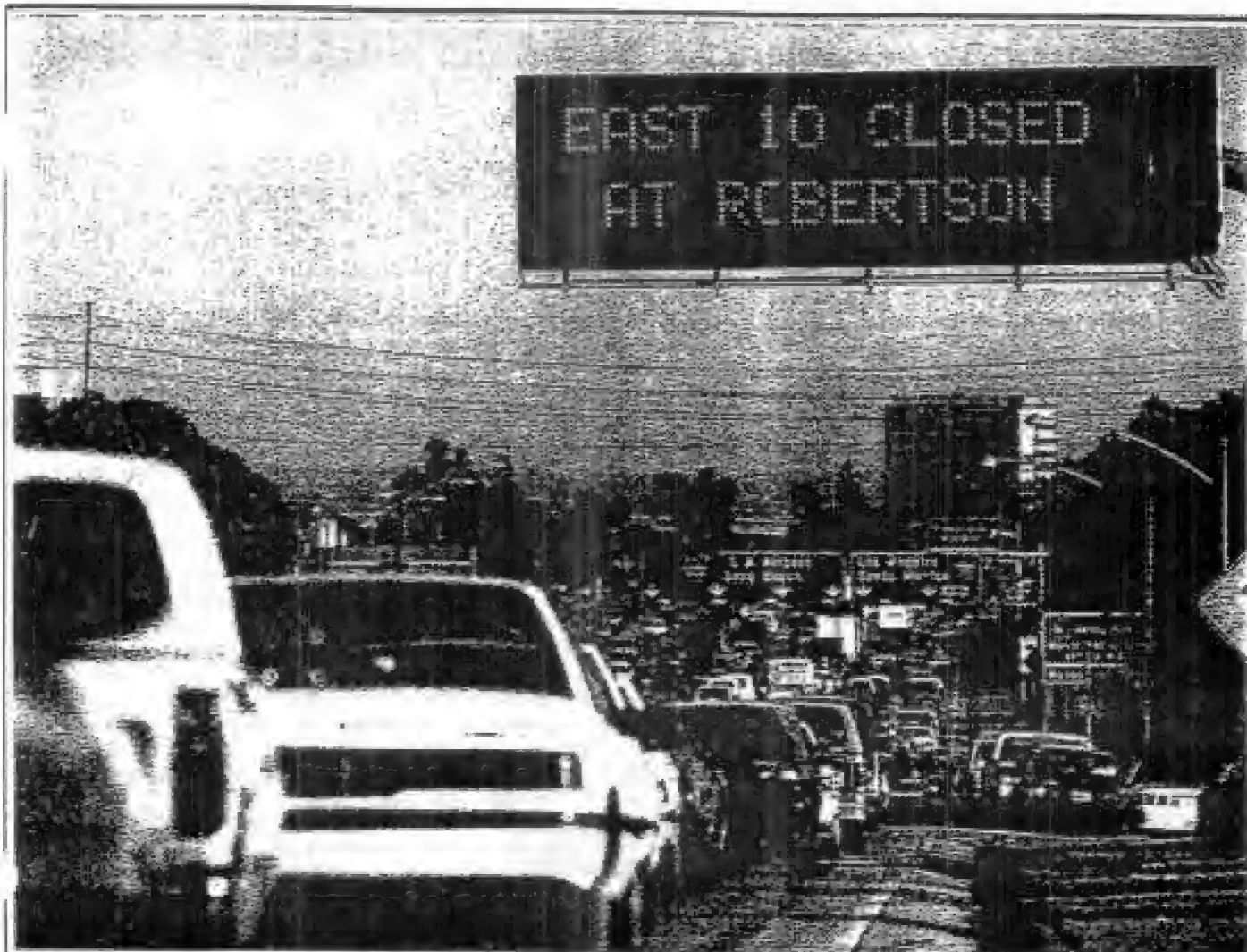
Prices in the communications sector have also likely fallen much more sharply than the government numbers show. According to the Bureau of Labor Statistics, the producer price index for interstate telephone service has risen by 2.4% over the past five years. Yet this figure doesn't take into account the discount calling plans that most long-distance companies now offer. Nor does it

adequately track the cost and use of leased lines. The BLS hopes to remedy some of these problems with a new index for telephone prices, perhaps by January.

The information economy also has a much larger productive capacity than the current government statistics indicate. For the moment, the main measure of how close the economy is

TELE-SELLATHON
HOME SHOPPING
CLUB HAS ADDED
THOUSANDS OF
JOBS IN NINE YEARS





to its maximum operating rate is the Federal Reserve's industrial capacity utilization number. While this includes utilities that sell electricity and natural gas, it leaves out telecommunications. That means there is no good measure of the amount of spare capacity in the U.S. telecom network. That's an important omission, since many businesses have become increasingly dependent on reliable—and widely available—communications services.

Even the investment boom of the past few years understates the true value of the spending on information technology. According to Commerce Dept. figures, investment in communications equipment has barely risen since 1990. What these numbers don't say is that for the same price, companies have been able to buy vastly more sophisticated switching gear and other telecommunications equipment, with new capabilities such as call forwarding.

Beyond those hidden by the measurement problems, there are some fundamental differences between the information economy and its predecessors. In the past, technological improvements such as railroads, auto plants, and steel mills required vast amounts of capital.

HONK IF YOU'RE ON-LINE AS TELECOMMUNICATIONS IMPROVE IN RURAL AREAS, URBAN TIE-UPS COULD GO THE WAY OF THE EDESEL

But because the price of information technology continues to drop so quickly, companies can spend less to get healthy improvements in productivity and quality. Indeed, in recent years, the productivity of capital—defined as the amount of output produced per dollar of plant and equipment—has gone up for the first time in the postwar era. "As the U.S. becomes an information-oriented economy," says William Sterling, an economist at Merrill Lynch & Co., "you may have less need for capital than you have in the past."

For example, phone companies are able to boost the carrying capacity of their existing fiber-optic cables by simply upgrading the electronics at either end. That means they can add to capacity without having to go through the expensive process of digging up old cables and installing new ones.

Even connecting all of the nation's homes to the Information Superhighway may cost less than expected. In Cal-

ifornia, Pacific Telesis Group and AT&T are estimating that it will cost an average of \$800 to wire each of 1.5 million homes with a combined fiber-optic/coaxial cable network that can carry the most advanced services. That compares with \$1,600 for the electronics and labor needed to run a fiber cable all the way to the home. "The fiber-only estimates were scaring everybody off," says Robert Clark, vice-president for marketing and sales at AT&T Network Systems. "We've been able to see another way of getting all the services."

If these lower estimates turn out to be right, it won't come as a total surprise. On a comparable basis, the price of information-technology equipment has dropped by 23% over the past five years, according to Commerce Dept. numbers. This trend, if it continues, will have important implications for interest rates. If companies need to borrow less money to finance their investment in high-tech equipment, that will keep overall rates lower than they would have been otherwise. And that will benefit homeowners, the government, and other borrowers.

Still, there's the matter of those losers from the shift to the information

economy. At the top of the list are the workers who have lost their jobs as companies reengineer their businesses. The reduction in staff can be enormous. At USAir Inc., 650 people were once needed in the revenue accounting department. Now that much of the process has been automated, only 350 people are needed to do the work, says Senior Vice-President and CFO John W. Harper. And at many companies, the downsizing isn't over. "Where will all these people be employed?" asks Lester Thurow, an economics professor at MIT and former dean of the university's Sloan School of Management. "It's not at all obvious."

Ironically, some of the biggest staff reductions have come at computer and telephone companies, which are at the heart of the information economy. Competitive pressures play a role, but these

cuts are being driven mainly by technological advances that let the phone companies, for example, do with fewer operators, maintenance people, and other workers. NYNEX Corp., which supplies local phone service in New England and

New York, announced plans last January to pare its workforce by 22%, or 17,000 people, by the end of 1996. Overall, employment in the telephone and computer manufacturing industries has already dropped by 154,000 since 1988, with more cuts to come.

Also at sea in the information economy are unskilled workers and people who can't keep up with technology. Indeed, recent studies suggest a hefty payoff for workers who feel at home in the digital world. Princeton University economist Alan B. Krueger estimates that people who use computers at work earn 10% to 15% more than

colleagues in similar occupations who do not use computers. Says Lawrence Katz, chief economist at the Labor Dept.: "There is very strong evidence that people who work with computers earn higher wages."

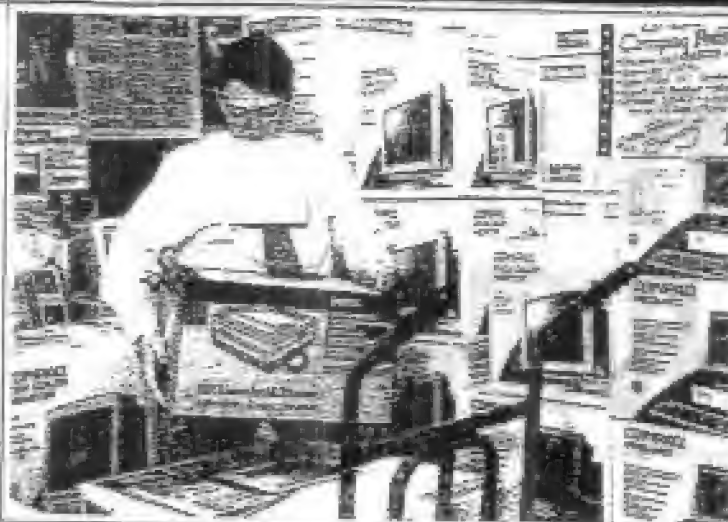
Still, even if some people are being left behind, the information economy is creating thousands of new businesses and jobs. For example, the Home Shopping Network—which sells jewelry and other merchandise on cable TV—has grown to employ some 5,000 people, up from 600 in 1985. At the other end of the spectrum are startups such as SandPoint Corp., a Cambridge (Mass.) maker of software that helps people track down information in databases. Over the past year, SandPoint has grown from 15 to 32 employees, and it's still expanding. Overall, the number of jobs in the software, data processing, and information retrieval industries has risen by 31% since 1988, and these industries now employ more people than the auto industry.

Besides creating jobs, the information economy may even make it a bit easier

**Now, rural South
Dakota schools can
offer Spanish via
interactive TV**

to match workers to existing jobs. The Online Career Center, based in Indianapolis, provides job and résumé listings on the Internet. Since it went online in June, 1993, observes Director William Warren, it has become one of the most popular databases on the system, with 13,000 to 14,000 job openings listed and nearly as many résumés. Ultimately, nationwide listing services such as this could make labor markets more efficient and help lower unemployment.

The effects of the information economy are even reaching into rural areas by shifting development away from congested urban regions. With more and more parts of the country having access to high-capacity telecommunications, companies can now put jobs such as order-taking in remote locations without losing touch with the rest of the business. "What telecommunications allows



NEW ERA FOR EXPORTS: UNLIKE SERVICES, AMERICAN COMPUTERS AND SOFTWARE CAN BE SOLD EASILY OVERSEAS

you to do is put the right facilities with the right labor," notes Ken Kuhl, a consultant with Moran, Stahl & Boyer, a business relocation firm.

Technological advances will have an even more profound impact on the vitality

of rural areas by bringing big-city services and amenities to small towns. For example, the telecommunications network operated by the state of South Dakota enables rural schools to offer Spanish classes via interactive TV—something they would never have been able to do on their own. The information revolution, says South Dakota Governor Walter D. Miller, "is going to change the face of

South Dakota as much as rural electrification did."

That's an apt parallel. Just as the U. S. economy today would be unthinkable without electricity, so will tomorrow's economy be spurred by the free flow of information. Judging by the explosive growth of information technology so far, the juice is only starting to flow.

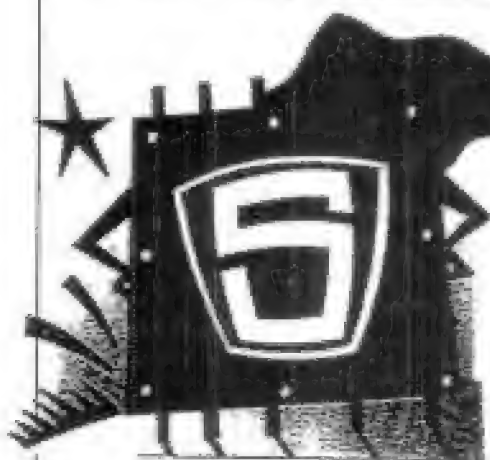
With Ira Sager in New York, Howard Glickman in Washington, and bureau reports.

THE KEYS TO THE FUTURE



Nothing is final. Better mousetraps come along all the time—not in one neat package, necessarily, but bit by bit. Here's the shape of things to come: The 10 critical technologies of tomorrow

HARDWARE



Semiconductors

For 20 years, the Information Revolution has been built on sand—silicon-based chips, that is. Each year, by cramming more circuits on a memory or microprocessor chip, engineers have made computing technology cheaper, more plentiful, and more adaptable to new uses. Without those annual improvements, the spread of information technology would slow, and new appli-

cations just slightly beyond our grasp—accurate weather forecasts, interactive TV, or computers that understand whatever you say, for example—might not be practical.

Until recently, scientists had nagging doubts about the future of silicon. They feared that by around 2000, they would run up against physical barriers—a size below which silicon transistors couldn't work. That would require shifting to other, more costly materials. But that supposed size limit has now been lifted by new research at AT&T Bell Laboratories, Hitachi, NEC, Toshiba, and other labs. "As far as we can see," says Paul M. Horn, IBM's director of silicon technology research, "there's no science limit for the next 30 years."

In other words, chipmakers will continue doing what they have always done: divining ways to scoop out ever-tinier trenches for the pipelines that carry data around silicon real estate. The more plumbing that's buried in each silicon plot, the

more magnificent the edifices that electronics engineers can erect for crunching numbers, routing telecommunications traffic, or presenting a friendly face to users. Smaller is thus always better—yet never good enough.

The progress so far has been astounding. When Intel Corp. developed the microprocessor in 1971, the year after it invented the dynamic random-access memory (DRAM) chip, the state of the chipmakers' art enabled engineers to lay down lines that were 6.5 microns (millionths of an inch) wide. That yielded 2,300 transistors on a chip the size of a thumbtack. Memory chips held 1,024 bits, and microprocessors were capable of slowly crunching eight bits of data at a time. Seven technology generations later, chipmakers are etching circuits that are just 0.5 microns across, making it possible to cram up to 35 million transistors on a chip. The result: DRAMs that can store 16 million bits of data and 64-bit microprocessors that are 550 times as powerful as the first Intel chip, or about the speed of a 1986-vintage IBM 3090 mainframe.

What's in store? A continuing exponential growth in processing speed and storage capacity. In reducing line width by 50%, the number of possible transistors doesn't just double—it jumps more than tenfold. And as the circuitry gets more dense, you get another performance boost. With everything crunched

PUSHING THE STATE OF THE ART

	1973	1984	1989	2002
SMALLEST LINE WIDTH (MICRON)	0.5	0.35	0.25	0.18
DRAM CAPACITY (MEGABITS)	16	64	256	1,024
MICROPROCESSOR SPEED (MEGAHERTZ)	150	100	400	500+

SOURCE: SEMICONDUCTOR INDUSTRY ASSOCIATION & COMPANY REPORTS

closer together, signals can zip more quickly between transistors, so the microprocessor's heartbeat, or clock speed, can be increased. Just a couple of years ago, the microprocessors in most personal computers ran at 25 megahertz. Now, Intel Pentium chips run at 100 mhz, while Digital Equipment Corp.'s speediest Alpha chip cruises at 190 mhz. And for reduced instruction-set computing (RISC) chips such as Alpha or the IBM/Motorola PowerPC, speeds could hit 400 mhz by decade's end and 500-plus mhz in 2002.

For the foreseeable future, at least, it seems that silicon will be able to match the needs of an increasingly demanding market. We'll move from today's 16-megabit DRAM chips to 64 megabits, and then, around 1999, to 256. And the first next-century memory chips will store an incredible one billion bits of data—eight such chips could hold the *Encyclopedia Britannica*. Applying this chipmaking technology to microprocessors will yield chips as powerful as an entry-level Cray 3 supercomputer from Cray Research Inc. But the chip will cost a few hundred dollars, not a few million. So whatever tasks get dreamed up for tomorrow's computers, silicon has the horsepower.

By Ods Part in New York



Optoelectronics

Call it the invisible backbone of the Information Age. Every melody from a CD player, every voice over long-distance phone lines, each page from a laser printer comes to you courtesy of a felicitous marriage of light and electricity known as optoelectronics. Without this underlying technology, "there wouldn't be an information infrastructure," says Edmond J. Thomas, execu-

tive vice-president for science and technology at Nynex Corp.

Right now, fiber optics is making the biggest difference in telecommunications. Sophisticated lasers transform electrical representations of conversations, faxes, or data into pulses of light, which speed through the fiber to their destinations, where they are converted back into electricity. Today's most capable fiber telephone lines can carry a gigabit or two of information—roughly an *Encyclopedia Britannica*—in a second. That's a 10,000-fold improvement on copper.

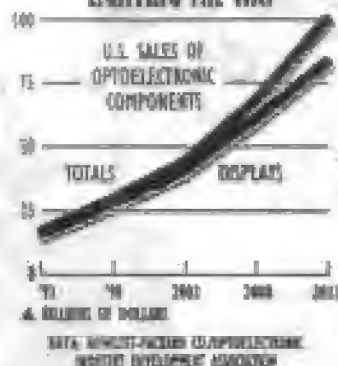
The total market for optoelectronic components is relatively small, now consisting mostly of flat-panel displays in laptop computers. But without components such as a \$2 laser in a CD player, entire categories of products would be impossible to build. That's why the U.S. Optoelectronics Industry Development Assn. (OIDA) says the technology is responsible for markets worth \$50 billion today—and more than \$200 billion within a decade.

Researchers continue to push the optoelectronic envelope. By recording information as holograms or using so-called blue lasers with shorter wavelengths to read more tiny "spots" on a disk, scientists may create devices with unheard-of storage capacity. Scientists now predict they'll be able to pack perhaps 18 trillion bits of data on a single 12-inch platter (page 63). With new lasers and optical switches, they're pumping data over fiber at 10 gigabits per second and figure on hitting 100.

What this could mean is an explosion of low-cost, high-speed communications capacity and the ability to store floods of digitized video and sound—two key components of the Information Superhighway. But there's a catch: cost. "Much of the current effort is aimed at driving the cost down—in some cases, down two orders of magnitude [by a factor of 100]," says Davis Hartman, head of an optical interconnection research group at Motorola Inc.

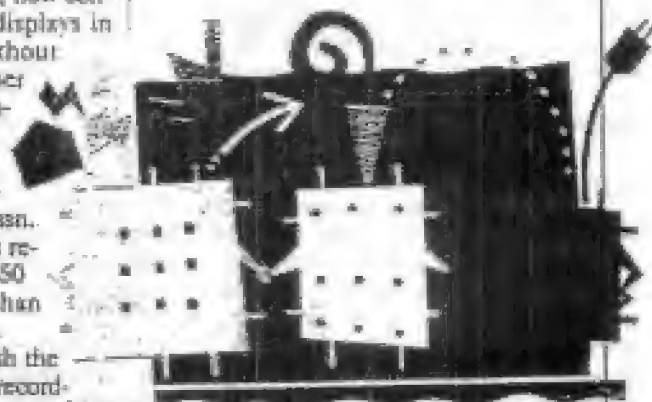
Breakthroughs in manufacturing would help. One possibility, being developed by startup Photonics Research Inc. in Longmont, Colo., is fashioning

LIGHTING THE WAY



entire arrays of hundreds of lasers on a single wafer, instead of individual devices. Such arrays not only cut costs greatly but also offer better performance in everything from printers to high-capacity networks. At the Naval Research Laboratory, researchers have etched patterns directly onto optical fibers as the glass is being made. The result is an inexpensive optical sensor that can be threaded through plane wings or skyscraper walls to warn of stresses and strains. With such advances on the horizon, the role of fiber optics will become anything but invisible.

By John Garry in Washington



Parallel Processing

You and 10 friends could paint your house a lot faster than you could paint it by yourself, right? If you understand the logic of teaming up to compress the time it takes to get a job done, you've already got the basic idea behind parallel processing, the key technology pushing the power curve on upcoming generations of large-scale computers.

The appetite for computer power remains boundless. Today, the digitalization that is converting TV, movies, magazines, and phone calls into the 1s and 0s of computers is putting undreamed-of demands on computing hardware. Even the most powerful supercomputer processor alone is no match for the job. Which is why parallel processing will be a critical technology in making multimedia and the Information Superhighway move from hype to reality.

In parallel processing, computer archi-

texts achieve Superhighway speeds by lashing together anywhere from a couple to many hundred processors and programming them to work in concert. Software divvies up tasks among all the processors—much as you might ask Ed to paint the porch while Sue starts on the window frames. The trouble is coordinating those processors—making sure that the right processor gets the right piece of information at the right time—is tough.

But progress is being made. Just a few years ago, parallel processing was the province of scientists and engineers. Now it is moving into the mainstream, primarily for applications that require huge databases. Wall Street firms are using parallel computers to tear through real-time data to evaluate financial instruments. Wal-Mart Stores, Hallmark Cards, and American Express are tracking the daily spending patterns of thousands of consumers. And companies such as Bell Atlantic Corp. are using massively parallel computers to act as "video servers" as the companies begin to test video-on-demand.

In the not-too-distant future, parallel machines may be the only form of supercomputers, mainframes, or high-end network servers that survive. IBM recently brought out its first parallel-architecture mainframe and sells massively parallel machines for engineering. Other major computer makers—including Unisys, Hewlett-Packard, Silicon Graphics, and Sun Microsystems—are switching to parallel computers, too. "At least in advanced computing," says industry analyst Gary Smaby of the Smaby Group, "there won't be anything that isn't parallel processing in the next three to five years."

Eventually, parallel computing will move to the desktop. Already, some organizations are using networks of workstations to simulate one giant parallel machine, running big database programs or crunching scientific equations when they are idled at night. And single PCs may need parallel power to keep up with demanding applications, such as speech recognition. Engineers at Intel Corp. say there are no hardware barriers to linking multiple microprocessors on a single chip. That holds out the promise of an intriguing technological oxymoron—a single/multiprocessor. Sounds weird, but it could give computers the power to keep the Information Revolution churning.

By Russell Mitchell in San Francisco



Storage

Can conventional disk drives—the devices that have been keeping data on magnetic platters for the past 20 years—keep up with the demands of the Information Superhighway? After all, a single digitized movie takes about 2 billion bytes to store, and the typical personal-computer hard disk stores only 210 megabytes.

The answer: yes. If routine technical improvements stay on track, by 1996 it will be common to have inexpensive hard-disk drives with a billion bytes of capacity in desktop PCs. After that, there will be an extra boost in bytes from ultra-sensitive recording heads that use an electrical phenomenon called magnetoresistance (MR) to pack data more densely on the disk. MR—so far made only by IBM—has doubled the pace of storage capacity improvements since 1992. Now, most drivemakers are experimenting with "giant MR," which could boost storage density thirtyfold by the year 2000.

Mechanics and materials are improving, too. More precise positioning mechanisms will allow heads to fly closer to the disk surface to read denser data. Drivemakers such as Seagate Technology Inc. are trying out ceramic and glass disks, which are harder and flatter—so more can fit in one drive disk stack.

Storing multimedia images, as well as spreadsheets and e-mail, presents special challenges. As drives warm up, they interrupt data flows briefly to adjust recording-head position. Nobody notices such delays with conventional data. But interrupting continuous streams of video means blurry pictures. So Quantum Corp. and Hewlett-Packard Co. have designed drives that calibrate heads with no delays.

Diskmakers have other tricks, too. Faster controller chips and a new digital formatting scheme will team up to move data faster from disk to screen. The data may come from multiple drives in one big box, called disk arrays. These devices are popular in data-processing shops and can economically store the data needed for instant video-on-demand.

Back on your desktop computer, big improvements in CD-ROM drives are in the works. First, Japanese component makers are working on lighter laser heads that can be moved more quickly across a disk. An even bigger boost will come from spinning disks faster—to enable a CD-ROM to handle up to six times the 150 kilobytes a second the basic CD churns out now. Also on tap: lasers using narrow blue beams to read denser data than red lasers can handle (page 61).

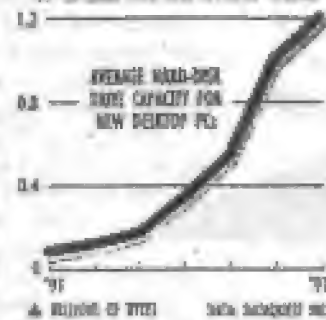
In short, diskmakers are determined to prove that their technology has a lot of life left. Says IBM Vice-President Christopher Bajorek: "We're probably decades away from any fundamental obstacles that would inhibit the progress of these technologies."

Indeed, even in the age of video-on-demand, disks will remain the workhorses of the Information Revolution. And other forms of storage won't go away, either. When a customer or-

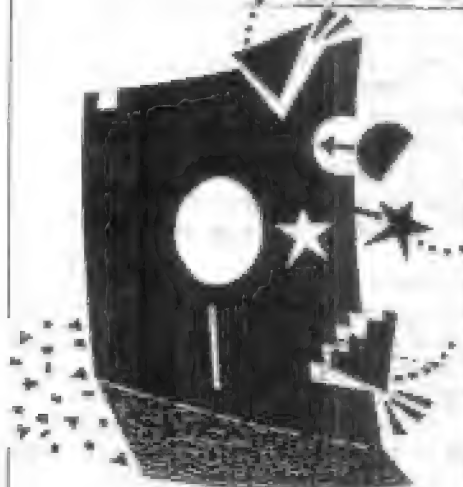
ders up a movie, it will be copied from disk arrays onto high-speed semiconductors. When the movie isn't likely to be viewed often, it will be erased from the disk and transferred to tape. Predicts Dataquest Inc. analyst J. Philip Devin: "The whole storage industry is going to thrive." More important, so should its customers.

By Robert D. Heif in San Francisco, with Neil Gross in Tokyo

A GIGABYTE ON EVERY DESK



SOFTWARE



Object Programming

When "object technology" emerged in the 1980s, experts hailed it as a way to make programmers more productive, speeding new applications to market and finally getting corporate computer departments caught up on their backlogs. That's a pretty daunting order for a fledgling technology. Now, experts are hoping for something more: They're figuring that object software can play a pivotal role in making incredibly complex information networks manageable and usable.

ming and data, rather than separating them as in conventional programming. So an object called "customer X" would include not only all data about the customer but also some computer code for communicating with other objects. That way it can respond when an object called "marketing survey," asks for data on customers. And these objects can also work across networks. So a manufacturer in England looking for a better flame-retardant plastic might be able to send an object around a network to talk to other objects at research labs in California to find a suitable material.

As information networks spread and as more special-purpose objects are created, all the computers on the network gain new powers—powers that will make it far simpler for people to find information and do business electronically. A searcher object, asked to find the gross national product of Peru, would relentlessly scour the network until it came to an object that "knew" the answer. The same technique might make it easy for a chief executive to glean important facts about a client or competitor. "Object programming is an enabling technology for creating smart

software that can sift through the tremendous amount of information out there," says Rick Jackson, director of product marketing at NeXT Computer Inc. Before object software can do all that, however, software makers have to work out a

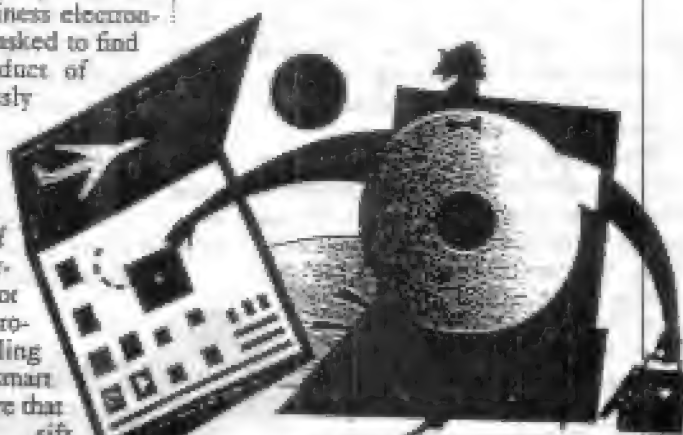
standard way of managing all of the objects on a network.

Software makers are beginning their move to object programming. Both Microsoft and Taligent, a joint venture of IBM, Apple Computer, and Hewlett-Packard, are developing object-based operating systems, expected sometime next year. Rather than slowly adopting object-programming techniques, the industry "should make a complete change to object-oriented software," says Mike

Potzel, vice-president for technical development at Taligent.

Why? Not only can the new technology speed you to conventional computer information—documents, reports, databases, and so on—but it is far more practical for managing the pictures, video, and sound of multimedia, all of which can become objects. Another huge benefit: Objects promise to eliminate the need to shift from one program to another when you want to switch tasks. Each function—writing or calculating, say—would be an object, like a hammer or a screwdriver, that you would pick up as needed. That will erase the lines between applications—and the barriers to sharing information among them. That in itself would be a revolution: computers that work the way we do, instead of requiring us to learn their arcane ways.

By Fred Gutzel in New York



Agents & Artificial Life

What is life? Scientists and philosophers through the ages have tried to answer that question by listing what seems to be unique to living organisms. Living things ingest food, for example, grow, reproduce, and finally die.

Now consider a computer. Can you create a program that embodies all of the definitions of "alive"? Researchers in the field of artificial life are doing just that. Their efforts could lay the groundwork for a new era of amazingly useful information systems. Across the world, programmers are creating artificial organisms from the primordial soup of digital bits that pulse through silicon chips. They float invisibly across the seas of computer networks, feeding on data, mating (passing on the characteristics of both parents to their offspring), grow-

THE OBJECTIVES OF OBJECT PROGRAMS

- ▶ Make program-writing faster and software more reliable by using pre-fab building blocks.
- ▶ Let different applications share common functions.
- ▶ Mix and match objects to customize.
- ▶ Break down barriers between different applications and types of computers.

The idea behind object technology is to break computer programs up into neat packages, called objects, which then serve as building blocks for larger programs. With programs made of objects, programmers are free to borrow objects from other programs or purchase them, rather than reinventing the wheel every time. To add functions, they can simply add new objects.

A huge potential payoff from object technology stems from binding program-

ing, learning, evolving, and even dying when their utility has passed.

Already, the techniques being learned by A-lifers are changing the way programmers build software. Rather than writing programs, A-life researchers unleash "genetic algorithms," strings of computer code that automatically generate new code and can combine like

Conventional robots are getting new programs from artificial life, too. Rodney Brooks, a professor at MIT, is programming robots with the instincts of an ant, so they wander freely, feeling their way around obstacles or, if blocked, retracing their steps. Researchers in Japan are working on programs that can travel the Internet, following the sunset around the globe to perform their tasks on idle computers after business hours.

Some researchers are also exploring whole populations of programs. Langton has a Swarm Simulation System, in which agents learn to interact. Just as an ant colony seems to show more intelligent be-

havior than an individual ant, Langton hopes software swarms will take on more complex behaviors.

Can such programs be considered "alive"? The debate rages. One definition of alive, offered at an A-life conference: "that which dies when you stamp on it." Alive or not, A-life could soon become a tool in our lives.

By Richard Brandt in San Francisco

THE FRUITS OF "ARTIFICIAL LIFE"

SOFTWARE "AGENTS" may be able to act autonomously, learning how to solve problems

SOFTWARE CODE that automatically evolves using a "genetic algorithm"

COMPLEX COMPUTER SIMULATIONS will predict environmental, social, or biological trends

ROBOTS programmed to mimic the simple reasoning of insects may "learn" to find their way

genes in an organism. They may evolve through random "mutations" induced by their creator or by "mating" with other successful programs. Unsuccessful offspring are killed off.

Researchers believe such programs are the most efficient way to solve problems with a huge number of variables. "If you can turn evolution loose on these problems, you might find solutions you wouldn't have thought of otherwise," says Christopher G. Langton, director of the artificial life program at the Santa Fe Institute, a private research group. At the Tokyo Institute of Technology, researchers are building genetic algorithms that may learn to schedule hundreds of processes in a factory. Supercomputer maker Thinking Machines Corp. is testing a genetic algorithm program called StarGene to sift thousands of pieces of data on millions of credit-card users to predict how the cards will be used. Several researchers are nurturing A-life programs to predict swings in the stock market.

A-life research also promises advanced software "agents," handy little programs that someday will sit in your computer and assist with tasks such as scheduling meetings or screening e-mail. Patti Maes, a researcher at Massachusetts Institute of Technology's Media Lab, is working on agents called "softbots" or "nobots." An e-mail softbot might spot patterns in the way you screen your e-mail and encode the routine in software, say putting memos from the boss on top.



Speech Recognition

Wouldn't it be great to deal with a computer on your own terms—say, by talking to it? That has been the dream of computer scientists for decades.

Continuous, speaker-independent speech recognition—where you could walk up to any computer and have it do your bidding, just like on *Star Trek*—still

remains elusive. Today, software that helps computers recognize any person's voice—speaker independence—is limited in vocabulary. Computers can understand your "yes" or "no" when you're asked if you wish to accept charges for a collect call. They can help you program your VCR. But programs with big vocabularies, such as Dragon Systems Inc.'s 60,000-word dictation package, must be "trained" for each user and require the speaker to pause after each word.

With every improvement in microprocessing power, however, scientists come closer to full speech recognition. Since increased memory and faster processing let computers handle more variables, new systems don't need to be trained for every speaker. And imaginative software tricks, such as using context to predict the next word in a given sequence, are improving accuracy. "Context alone has made an immature technology mature for some applications," says speech researcher Kai-Fu Lee, Apple Computer Inc.'s director of interactive media.

At IBM and elsewhere, scientists are exploring programs that enable computers to expand their vocabularies through experience—figuring out, for example, when "bad" comes to mean something good. Another promising approach is to enable computers to discern inflection, as when a speaker's voice rises to signal a question.

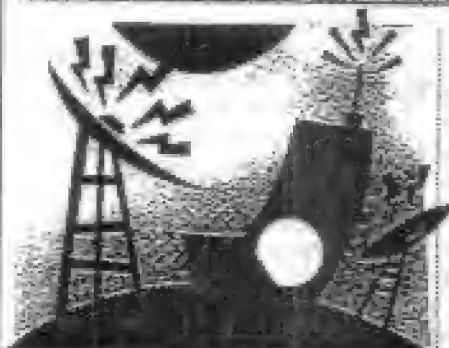
A shortcut is to restrict the vocabulary according to topic, such as travel, weather, or sports. Fidelity Investment Co., for example, is evaluating speech systems that will let customers call in and ask the computer for quotes on mutual-fund prices, rather than tapping in codes on a Touch-Tone phone.

When will speech recognition be reliable and economical? At the current rate of progress, researchers say it will be another decade

before speech recognition replaces the keyboard for most uses. Until then, it's unlikely that the Information Revolution will reach all citizens. "The only way that's going to happen is for computers to learn to understand what people say," says George R. Doddington, the Advanced Research Projects Agency official who distributes government money for speech research. Who knows? A few kind words, and computers might really become personal.

By Gary McWilliams in Boston

COMMUNICATIONS



Wireless

The promises of the wireless communications revolution are vast but can be summed up easily: high-quality voice and data service anytime, anywhere. Sounds good. But if you've tried calling at peak hours in Los Angeles or New York, you know today's cellular systems can't even guarantee consistent connections. So how is the brave new wireless world ever going to come about?

Technology, of course. Advances over the next decade promise to make wireless communications networks as capacious and reliable as fiber-optic lines. Within three or four years, wireless data could be transmitted at current wireline rates—double today's wireless data speeds—through the use of improved software and chip technology, particularly digital technology that will allow better compression.

Helping to prod the \$10.9 billion cellular business into the digital age is an ambitious throng of newcomers that hope to wrest big chunks of the wireless market from such well-established giants as McCaw Cellular Communications, Sprint, and the Baby Bells. These upstart networks—personal communications services (PCS), enhanced specialized mobile radios (ESMRs), and satellite-based setups—are pushing wireless technology to new limits. Says Sprint's Cellular Vice-President Keith Puglione: "The basic question is how much of the wireless market each will garner."

It will take several years for any of the new networks to challenge cellular. The most ambitious, the Teledeco Corp. satellite venture being funded by McCaw and Microsoft Corp. Chairman William H. Gates III, won't be ready until 2001.

In the meantime, cellular operators are moving slowly to digital to expand capacity and improve quality. One route is Time Division Multiple Access (TDMA), a technique that splits a channel into three time slots to handle three calls at once, compared with one call per channel on analog. McCaw has TDMA equipment in place in New York and other cities.

But most cellular companies are looking for a bigger punch by using Code Division Multiple Access (CDMA). It promises a 10- to 15-fold capacity improvement using a technology called "spread spectrum" that distributes a digitized message across a wide range of frequencies. On the receiving end, the phone reassembles the signal. Irwin M. Jacobs, president of CDMA developer Qualcomm Inc., says the technology will be widely available in 1995.

More immediately, cellular companies are beefing up the ability of their networks to handle data so that customers with wireless modems in notebook PCs can receive e-mail and other messages. McCaw has begun rolling out a system developed by IBM called cellular digital packet data (CDPD), which transmits "pockets" of data through sparsely used radio channels or during gaps in conversation. But CDPD works best for short files that can be sent quickly.

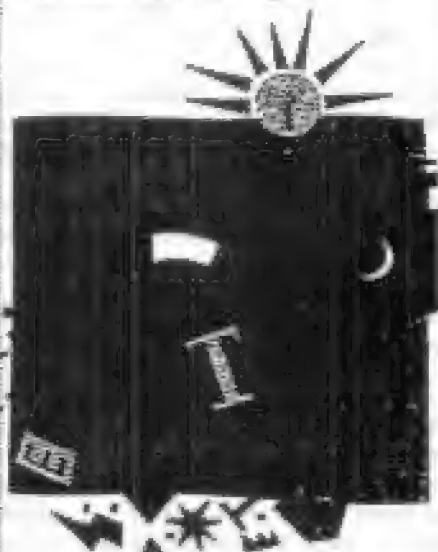
By the time cellular operators have CDMA systems in place, they are likely to face new competition from ESMR setups. These networks, using frequencies allocated for radio dispatch, already handle digital voice, data, and paging—in a single receiver. Spearheaded by startups such as Nextel Communications, CeeCall, and Dial Page, ESMR networks are now operating in only a few cities, however they are expected to go nationwide by 1996.

But the biggest threat to the cellular order is likely to be

PCS. A variation on cellular technology, PCS divides an area into many "mini-cells." That means that handsets to send and receive data and voice can be smaller and cheaper. PCS operators say their systems will work indoors and out so a single PCS handset will be your home and mobile phone.

The ultimate wireless advance will be truly unlimited communications. Motorola Inc.'s multibillion-dollar Iridium project, using 66 satellites, will guarantee customers willing to pay a stiff price the ability to call from any point on the planet. But by the time it's operating in 1998, it may amount to nothing more than "pie in the sky," says Ira Brodsky, president of DataComm Research in Wilmette, Ill. With all the advances in land-based wireless, the only place the "anytime, anywhere" promise won't easily be fulfilled may be the Gobi Desert.

By Kevin Kelly in Chicago

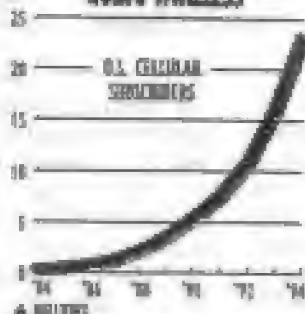


ATM Switches

We're all being conditioned to expect immediate gratification in the digital age—with a click of a button we'll summon Hollywood movies, stock quotes, electronic shopping services, or simply dial a phone call. But we may all be in for a wait—at least until a new switching technology, known as Asynchronous Transfer Mode (ATM), comes into widespread use. When it does, ATM's ability to funnel billions of bits to where they're needed "will make the network to end all networks," says analyst Paul D. Calahan of Forrester Research Inc.

ATM's trick? It divides the information into tidy packages, or cells, of 53

GOING WIRELESS



bytes each. The cells are coded, or stamped with an address, and then zipped over the network at high speed. A switch at the other end reads the coding and reassembles the phone call or e-mail message at the other end. The speed is amazing: AT&T says its ATM switch can pump 20 gigabits of data—or 1,600 copies of *Moby Dick*—every second.

Impressive. But relatively few corporations or telephone companies have ATM. The holdups: The lack of fully standardized equipment and software—so that all ATM brands can work together—and steep prices. Which means companies such as U S West Inc., a Baby Bell, are holding back. ATM would probably help U S West's planned interactive-TV trial. But Russ Skinner, director of video engineering for U S West Communications, doesn't believe that's the case: "Today, ATM is not mature enough or cost-effective enough to put in the network."

That could soon change. A 530-member industry consortium, called the ATM Forum, is hammering out standards. The two-year-old group is now tackling two critical issues: The first is a standard for emulating local-area networks that will let computer users link up with ATM networks; the second is congestion control—how to control data flow when enormous numbers of people attempt to call up the same material at the same time.

ATM Forum President Fred Sammarino predicts that a standard for the local-area-network emulation is just around the corner, while one for congestion control should be completed by early next year.

At the same time, ATM switch prices are falling. Today, switches for high-speed corporate systems and large phone networks can cost up to \$3 million, while switches for small networks cost \$1,500 per port, or user. Prices are still steep, but they've been cut in half each year since 1991 and are expected to keep falling as ATM becomes more broadly adopted.

In a few years, prices will likely fall to a "few hundred dollars" per user, says James Chiddix, senior vice-president of Time Warner Cable, which is launching an interactive-TV pilot in Orlando using ATM switching technology. Chiddix adds: "Today, ATM is limited. But that's about to change in an explosive way." When it does, the digital deluge can commence.

By Kathy Rebellio in San Francisco



Compression

Despite advances in semiconductors, disk storage, and optoelectronics, there may always be a struggle to accommodate all the electronic traffic and find a place to park digital information when it's not in use. And that will keep the pressure on compression—the science of squeezing reams of digital data into less space.

Advances in the mathematics of compression have already yielded impressive results. The Joint Photographic Experts Group's standard compresses still images into one-fourth the space. The Motion Picture Experts Group (MPEG) has agreed on two standards, MPEG1 and MPEG2, to compress video. They work by stripping out redundant information—say, the mountain in the background—between one frame and the next, so the computer only needs to deal with the information that changes. MPEG reduces the full video signal from 230 million bits per second to 1.5 million for VCR quality. Now, using MPEG and ultrasensitive electronic circuits, researchers are able to transmit four channels of TV over ordinary copper phone wires—an impossibility a few years back.

But experts say that progress in compression may be slowing. "We've already thrown away 99% of the video signal," says John Forrest, chief executive of National Telecommunications Ltd., a satellite-equipment manufacturer in Winchester, England. "Now, we have

reached a technological plateau." And moving beyond MPEG could be essential for making interactive TV and other advanced video services possible. The problem? Although decoding an MPEG signal is quick and cheap, encoding one is time-consuming and expensive.

There are some promising developments that could pay off—someday. Engineers have achieved impressive compression ratios using algorithms based on fractals, a branch of chaos theory. Fractals work well on images of landscapes and seascapes made up of recurring patterns, but for most video images, quality tends to suffer. And like MPEG, fractal images are costly to encode. "On balance, fractals don't appear to work any better than MPEG," says Jules A. Bellisio, executive director for video signal processing research at Bell Communications Research in Red Bank, N. J.

Wavelet theory is a more enticing alternative. Wavelet algorithms are very efficient at dividing the video image systematically into blocks and then describing each block with relatively concise mathematical equations. As a result, wavelet algorithms are as quick encoding an image as decoding it. And since they do not rely on predicting motion between frames as MPEG does, wavelet images tend to retain higher quality.

Still, wavelet images can have lines on the screen that reveal the image's blocklike structure, even in video that's termed "broadcast quality." The lines disappear only at compression ratios almost as low as MPEG's.

Another solution may lie in combining compression with more sophisticated display technologies. The human eye transmits vast visual information to the brain even though

the retina is a poor data pathway. Researchers suspect a video display with a web of thousands of tiny microcomputers—one for each picture element, or dot, of the video screen—might be able to generate good video images with far less data than conventional videos now require. Such a display might emerge in 3 or 10 years. If not, we could stall in traffic on the Information Superhighway.

By Fred Guterl in New York

THE BIG SQUEEZE

THE
CHALLENGE

THE
WAGE

UNCOMPRESSED

1:1

1:1

CURRENT METHODS

12:5

4:1

WAVELETS

63:1

20:1

* MPEG-1, MPEG-2, H.261, H.263, H.264, H.265



ROOM WITH
A VIEW:
GTE'S
NETWORK
OPERATION
CENTER

THE GREAT EQUALIZER



*Information
power is
getting cheaper—
and not only big
business will benefit*

BY IBA SAGER

When a truck rolls into the maintenance bay at Ryder System Inc.'s New Brunswick (N.J.) facility, all Karen Reinecke has to do is push a button to learn instantly what's ailing the vehicle. Reinecke, a technician for the \$4.2 billion transportation giant, simply touches the probe on the end of her handheld computer to a tiny coin-shaped disk on the truck's cab that has been gathering information on engine performance and fuel consumption from electronic sen-

sors under the hood. Gone is the guesswork—wrong 50% of the time—in finding engine problems. And with the sources of trouble more quickly identified, a truck's downtime can often be cut in half.

Information Age meets Road Warrior. Launched in mid-March, Ryder's Fast Truck Maintenance Service will capture every bit—and byte—of information on its trucks electronically. And thanks to all the new data, scheduling the com-

pany's 8,500 technicians and 162,000 trucks will be simpler, inventory tracking and parts ordering more efficient, and reports to fleet customers far more detailed. Better yet, Ryder will be able to use information it collects on engine-part wear to negotiate longer warranties from suppliers.

Chief Information Officer Dennis M. Klinger figures the \$33 million investment in new computer systems will pay for itself in two years. "Once we have that database of not only how to fix the truck, but of failures, we can predict which components act best in what applications—whether it's on the road or running around town supplying Burger King," he says.

Ryder's transformation is just one small part of a quiet revolution in the way business is viewing and using information. After years of big investments in technology to automate such tasks as order entry or billing, companies around the world are suddenly scrutinizing the information those systems capture to find easier and more efficient ways for their employees, customers, and suppliers to do business. The reason: Competitive pressure is pushing companies to downsize even as they improve both the goods they make and the service they provide.

Forced to do more with less, corporate giants such as IBM and AT&T are scurrying to reengineer their businesses by rethinking work flows and encouraging information sharing among once-autonomous fiefdoms such as purchasing, manufacturing, and marketing. But it's not just the big boys. Smaller companies and the self-employed are discovering that recent price drops and performance advances in PCs, wireless communications, and business software have given them cheap, potent weapons to compete gamely against big, deep-pocketed rivals (page 108).

Employees, too, are being buffered

by the Information Revolution. As businesses depend more on collecting, analyzing, and sharing information across their operations, they're demanding new worker skills for the Digital Age (page 112). A recent survey by compensation consultants N. E. Fried & Associates found that at least half the secretaries at most of 478 U.S. companies used be-

need it will be disadvantaged," says Robert M. Howe, a former Booz Allen & Hamilton Inc. consultant who since 1991 has run IBM's fledgling consulting business.

Luckily, a new generation of cheap computers and advances in software and networks allow U.S. business to ferret out information it couldn't afford to find in the past. Andersen Corp., for example, can manufacture windows to order because it installed PCs in stores to let customers configure, price, and order their own windows. And emerging technologies such as digital image processing and object-oriented software, which allow programs to be written in easy-to-reuse blocks, will let companies capture information without the cumbersome and costly process of translating it into arcane computer language. "All of a sudden," says AT&T CIO Ron J. Ponder, "it doesn't cost as much to have this technology."

That's why many companies are finding it's never too late to automate. With almost daily advances in computers, software, and networks, the cost of computing power is dropping roughly 30% every 12 months. Some companies that are only starting to go digital with networks of PCs or fast workstations find their processing costs actually lower than those of competitors who took the plunge in the early 1980s, when costly mainframes ruled the earth. In fact, USAir says it will get a leg up on competitors—and may spend half the amount rivals American Airlines and United Airlines pumped into their

mainframe computer systems. The reason: It's using hundreds of workstations to tear through critical ticketing information overnight—which some competitors take a couple of days to crunch.

Getting information faster and more accurately can dramatically alter the rules. Look at how the relationship between suppliers and customers has

How the Information Age Is Changing Business

The advance of digital technology is having a dramatic impact on businesses, their workers, and the suppliers and customers who trade with them. Here's how:

ORGANIZATION	New electronic systems are breaking down old corporate barriers, allowing critical information to be shared instantly across functional departments or product groups—and even with workers on the factory floor.
OPERATIONS	Manufacturers are using information technology to shrink cycle times, reduce defects, and cut waste. Likewise, service firms are using electronic data interchange to streamline ordering and communication with suppliers and customers.
STAFFING	New systems and processes have eliminated management layers and cut employment levels. Meanwhile, companies are using less costly computers and communication devices to create "virtual offices" from workers in far-flung locations.
NEW PRODUCTS	The information "feedback loop" is collapsing development cycles. Companies are electronically feeding customer and marketing comments to product-development teams so that they can rejuvenate product lines and target specific consumers.
CUSTOMER RELATIONS	No longer simply an "order entry" job, customer-service representatives are tapping into companywide databases to solve callers' demands instantly, from simple changes of address to billing adjustments.

sic spreadsheet software. And factory workers at world-class manufacturers such as Motorola Inc. must have the math and basic computer skills to run computer-programmed machinery or use statistical process controls to monitor quality on the production line. "The companies that do not provide information on an accessible basis to those that

THE NEW FACE OF BUSINESS

changed for retailers. In the 1970s, U.S. merchants started to replace clunky cash registers with electronic point-of-sale terminals that made tabulating receipts easier. But it wasn't until the 1980s, when scanners and bar codes became de rigueur, that retailers fully appreciated the wealth of information they have on their customers—everything from

TOKEN EFFORT: ELECTRONIC TOLLBOOTHS IN GEORGIA

you're at the [wrong] end of the chain."

Retailing is by no means the only industry in the midst of this info-tech revolution. Whether it's IBM giving salespeople once closely guarded data on product costs so that they can quickly respond to customer bids or Ryder's digitized garage, companies in almost every industry are keen to gain competitive advantage by employing digital technologies to manipulate information. "The more you know about your customers, the more you're able to predict

Klinger says the new technology his company is using is the only way it can boost its customer satisfaction rating to 95%, from the current 88%, by the end of 1995.

Edward L. Schrenk, senior vice-president at United Services Automobile Assn. (USAA) in San Antonio, goes further. He credits much of the insurance and financial services company's success to its commitment to technology. In just 15 years, USAA has mushroomed to the nation's fifth-largest private auto insurer

and fourth-biggest provider of homeowners' coverage, with \$18.5 billion in assets.

Since 1969, USAA has spent \$130 million on computer and imaging technologies to boost customer service and lower costs. Today, it boasts an information system so advanced that it can track minute details, such as which auto parts are getting fixed most often. Why bother? USAA passes that data to parts suppliers who then make those parts if there is a chance for improvement or if they can make them more

cheaply. The Big Three also get data from USAA to improve their parts.

Likewise, USAA had been trying for a long time to get glass shops to repair windows that had punctures outside the driver's field of vision, but no cracks. But shops would rather pocket the \$275 to replace an entire windshield than charge \$35 to repair it. So even though USAA offered to waive the deductible if customers would fix the glass, body shop owners were convincing drivers to replace the whole thing. Only when USAA started capturing data and publishing the repair record of various shops in its newsletter did this start to abate. The shops realized where they stood relative to the competition and didn't want to lose USAA's referrals. The percentage of repairs zoomed to 28% from 5% in just four years.

To be sure, not all companies have been as successful in making huge investments in technology pay off. The problem, consultants say, has been the traditional view of technology as a tool simply to cut costs and support already-



how often they use credit cards to what color socks sell best on Friday night.

Now, instead of taking product deliveries when vendors dictate, merchants such as Wal-Mart Stores Inc. increasingly are telling manufacturers what they want and exactly when and where they need it. Meanwhile, savvy suppliers are tapping this information to fine-tune their own production schedules according to what consumers are buying. "Point-of-sale [technology] changed the balance of power between retailers and consumer goods manufacturers," says Gerald Loev, former CIO of Prudential Insurance Co. and now head of a consulting service for Computer Sciences Corp.'s Index unit. Adds Deloitte & Touche Managing Director William Atkins: "If you're the vendor,

what they want, and the more you're able to deliver products that the customers want to buy," says John W. Harper, chief financial officer for USAir Inc. "In some cases, you can even create demand if you have the right information." That way, USAir can "micro-analyze" data on the 160,000 people it carries each day to find the best fare and schedule to fill its Friday afternoon flight from Pittsburgh to Harrisburg.

Above all, the companies most adept at exploiting technology to mine data are seeking something basic: to please customers. In fact, a recent Computer Sciences survey of information system managers at U.S. and European corporations found that customer service is the No. 1 focus of their companies' investments in technology. Ryder's

Something as
simple as sharing
an idea can pit
technology against
human nature

THE NEW FACE OF BUSINESS

existing operations. In the early 1980s, companies such as General Motors Corp. were convinced that huge investment in robotics to automate manufacturing plants would boost productivity. It didn't. "In the past, the technology was seen as a peripheral vehicle for implementing the strategy," says Joe Carter, managing director of Andersen Consulting's new technology center in Palo Alto, Calif. "Today, technology is the strategy."

In fact, much of the action in corporate information systems now involves capturing data on customers. "We have to have an end-to-end [direct] relationship with our customers," says AT&T's Ponder, who was recruited by the telecom giant less than a year ago to help link corporate goals with AT&T's massive technology investments.

For a \$67 billion behemoth like AT&T, understanding customer needs is critical. Ponder argues that large companies will find that future opportunities for growth will be in smaller markets or niches, with customers they already have. "You can't win anymore by selling 10 million of a product," he says.

The real breakthroughs will come from the ability to "mine" information—the process of quickly gathering and analyzing the millions of bits of data that a business generates each day—and steer various pieces to the right people within the organization. "Power used to be that you controlled the information," says IBM's Howe. "Now, power comes from providing greater access to the information."

But first, companies have to know their customers in even more detail. Dell Computer Corp. CIO Thomas L. Thomas says new computer systems he is installing will make Dell the "Nielsen" of the computer business. By using new systems to track customers' detailed buying habits, Dell hopes one day to be able to anticipate their needs and

call them with just the product they're most likely to buy—say, more memory or new software.

Dell is also investing heavily in a global information system that will, among other things, enable a registered repairperson in any country to pull up data on any Dell PC—in his or her native tongue. That does away with the need to print manuals in many different languages. Over time, Dell even plans to

rather than tapping Dell's service desk.

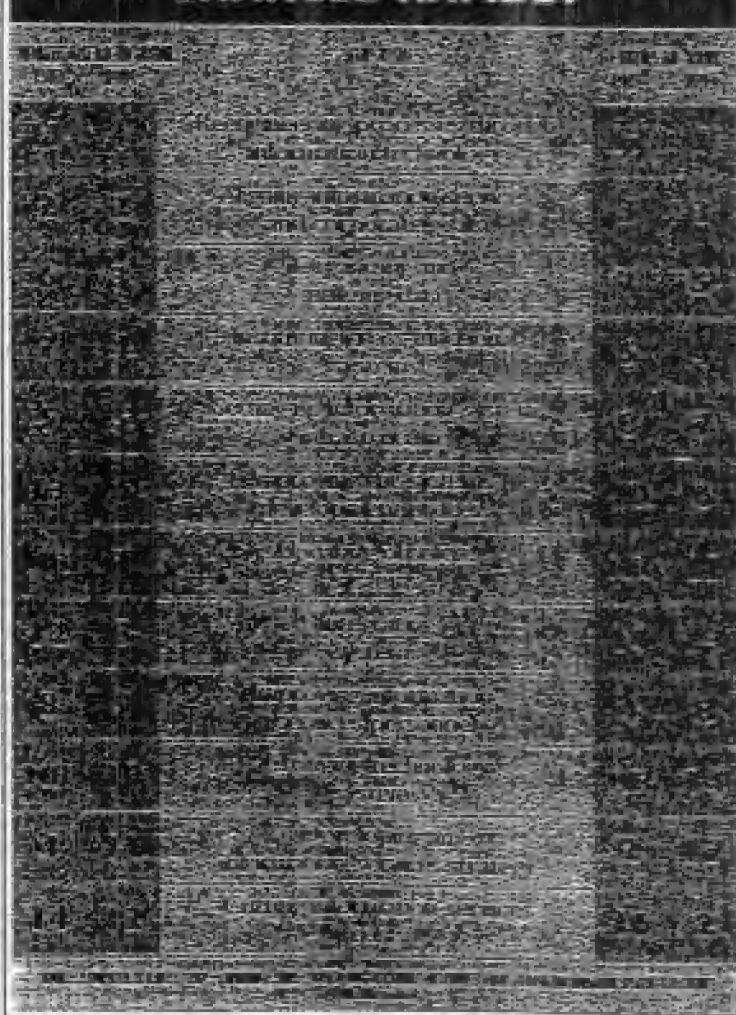
Indeed, customer service is one of technology investment's hottest areas. Good customer-service systems can cut field-service costs, improve relationships with existing customers, and eventually generate new sales. GTE Corp. customers used to get bounced around until they found the right department or person to send out a repairperson or process an order. That's because operators could do little more than take down basic information and pass the caller along. "It proved to be not very good customer service, but also very expensive to us," says Lanny Russell, a vice-president of GTE's telephone operations.

GTE decided that the best way to service its customers was to offer "one-touch" service—from the same operators who used to pass the call. That led the company to start developing a new system in 1992 to give operators access to several corporate databases quickly, allowing it instantly to send out a repair truck or even test the line electronically. In pilot tests, operators now solve problems with one call 35% of the time—compared with one in 200 just over a year ago. "We'll be able to provide better service in three years than we can today—and we'll be able to do it with much lower employee counts," predicts Russell.

Increasingly, companies are leveraging their existing information expertise into new businesses and markets. Dallas-based Amtech Corp. is a \$60

million company doing a thriving business supplying railroad companies with computerized systems that track exactly where railcars—and goods—are, anywhere in the nation. The company is using the same radio-frequency technology to branch into automated toll-collection systems to help state and local government reduce delays at toll plazas. With Amtech's system, commuters use a

INFORMATION PRIORITIES HAVE CHANGED RAPIDLY



give its more sophisticated customers access to this information system—allowing Dell to reserve more expensive human hand-holding for less savvy customers. Starting this year, some customers will be able to order additional PCs and receive shipment without ever talking to a salesperson. Later, they will be able to pull up technical information to diagnose and fix problems on their own,

credit-card-sized tag on their car that emits a signal at the tollgate identifying the owner without stopping. Monthly tabs can be settled via credit card.

For some companies, the only way to avoid falling behind competitors is to build organizations and systems that can adapt quickly to marketplace changes. At Texas Instruments Inc., the motto is "Change faster than change," says CIO J. R. "Bob" McLendon. TI, which

spends more than 4% of revenues on information technology, has long been a leader in the use of technology—it has had a sophisticated e-mail system for 15 years, for example.

Now, McLendon is using that expertise to build what he calls "the virtual factory," a system to let TI build any product any time at any of its factories worldwide—with all the engineering specs and invoices moving electronical-

ly. Already, TI's product designers can transmit designs and equipment setup instructions to automated manufacturing sites globally. For example, TI keeps its big million-dollar chip testers in the Philippines, but they're controlled by test engineers in Houston. While TI still has a way to go, McLendon boasts that the semiconductor group reduced cycle time from order to delivery an astounding 39% last year—largely by linking its

HOW TECHNOLOGY TRANSFORMS WORK

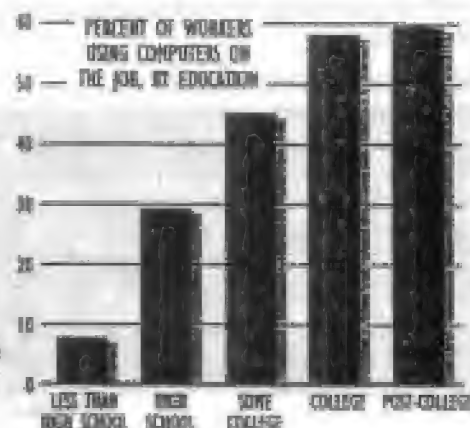
Innovation and technological change create winners and losers. Wal-Mart rises and Sears falls. Microsoft triumphs and IBM slumps. The same is true for labor: Some workers suffer job losses, while others get paid to ride the high-tech revolution. Right now, the squeeze on jobs is most obvious and worrisome. But over the past 200 years or so, there has been no long-term trend toward higher unemployment because of investment in new machines and technology.

SKILL AND
EDUCATION

REENGINEERING
THE WORKPLACE

FUTURE
RETURNS

Over the past two decades, the Information Revolution has been leaving the less skilled and less educated worker behind. But it has been a boon for those with at least some college education.



SOURCE: BUREAU OF ECONOMIC ANALYSIS

Service sector productivity is picking up smartly. One reason: High tech has taken root in service industries.

WORKERS USING COMPUTERS ON THE JOB, BY INDUSTRY

Finance, insurance, and real estate	71%
Public administration	62%
Transportation, communication, and other public utilities	40%
Services	39%
Manufacturing	36%
Mining	31%
Wholesale and retail trade	28%
Construction	13%

SOURCE: BUREAU OF ECONOMIC ANALYSIS

With technological leadership in U.S. hands, high-tech producers pay workers well.

AVERAGE WEEKLY WAGE FOR PRODUCTION WORKERS IN 1991

High-technology manufacturing industries*

\$565.75

All other manufacturing industries

\$474.23

*TRUCKS FROM 12 INDUSTRIAL GROUPS, INCLUDING INDUSTRIAL INORGANIC CHEMICALS, GASES, ENGINES AND TURBINES, COMPUTERS AND OFFICE EQUIPMENT, COMMUNICATIONS EQUIPMENT, ETC.

SOURCE: BUREAU OF LABOR STATISTICS

computers into one global network. Monitored from a space-age control center near Dallas, TI can reengineer processes on the fly—though the chips may be ordered in Japan, developed in Texas, and made in the Philippines.

If concepts such as the virtual factory or the virtual office are going to take off, however, companies will have to change their corporate cultures in addition to adding new digital technology.

As companies break down the barriers between departments and their customers, "sharing information becomes really critical to any organization's success," says Thomas H. Davenport Jr., a consultant with Ernst & Young.

The problem, Davenport argues, is that many companies spend big on technology to allow employees to share information, but forget that sharing ideas is an "unnatural act" in corporate cultures

that reward individual achievement. "If we really cared about information sharing, we would start to evaluate people by how well they share," says Davenport. With the amount of information that flies around organizations growing daily, focusing on information sharing—instead of just information technology—may be the next revolution.

With Peter Hurren in Dallas and bureau reports

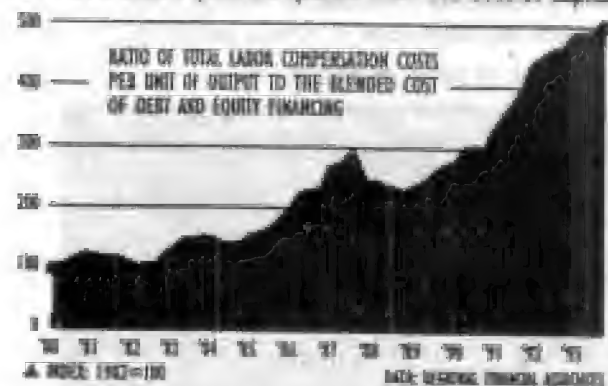
Technological change and office automation will shrink these jobs...

PERCENT EMPLOYMENT CHANGE, 1992-2005

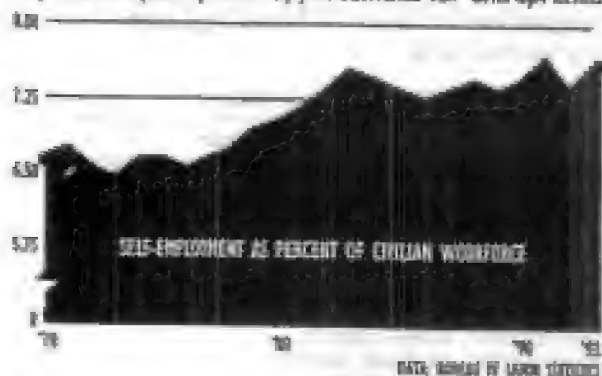
Computer operators	-39%
Billing, posting, and calculating machine operators	-29%
Telephone operators	-28%
Typists and word processors	-16%
Bank tellers	-4%

DATA: BUREAU OF LABOR STATISTICS

Business investment is skyrocketing as the cost of labor has more than tripled compared with the cost of capital.



Be your own boss. Computers linked to the Superhighway could open up new opportunities for entrepreneurs.



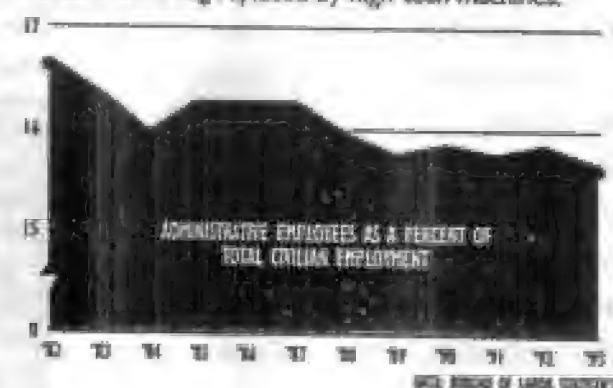
...but technology also generates new openings in the info-tech world.

FIVE FASTEST-GROWING OCCUPATIONS REQUIRING A COLLEGE DEGREE, 1992-2005

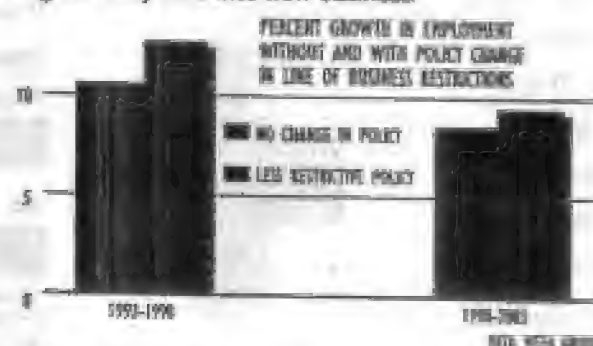
Computer engineers and scientists	112%
Systems analysts	110%
Physical therapists	88%
Special education teachers	74%
Operations research analysts	61%

DATA: BUREAU OF LABOR STATISTICS

Companies are doing more with less. Classic white-collar workers are being replaced by high-tech machines.



But the impact of the Superhighway on jobs depends on government policy. Look at the effect of permitting the Baby Bells into new business.



MISSION-CRITICAL VIEW

By ROBIN BLOOR

The Disappearing Programmer

AS OBJECT-ORIENTED CONCEPTS TAKE HOLD, THE NEED FOR SPECIALIZED PROGRAMMERS IS DECLINING.



Software is expensive. It might not seem so when you can acquire PC databases, spreadsheets, or word processors for less than a hundred dollars, but it is. PC software is cheap only because the market for it is very large. Producing the software itself, whether it is a manufacturing

system, an accounting system, or word-processing software, is an extremely labor-intensive process. Even with the most productive tools, the fastest workstations, and the best people, good software takes time (often several person-years) to produce.

Productivity is one of the biggest issues in the software industry, and so far we have not been good enough at providing it. It's true that the industry has provided 3GLs, 4GLs, CASE tools, databases, OO languages, automated testing software, and a whole host of other products to increase productivity exponentially every year since the 1950s, but it has not risen fast enough for comfort. For instance, it has not risen fast enough for major corporations to consider completely rewriting most of their mission-critical systems.

All the evidence suggests that the demand for software has exceeded the industry's ability to supply it. The key piece of evidence is the number of people whose jobs involve writing software. The number of programmers has been on the increase for decades.

Programmer Statistics

An interesting workshop was held in the U.K. last April, titled "Software 2000 — a View of the Future." Academics and commercial experts from the U.S. and Europe gathered together and tried to predict the future of the software industry, focusing on programmers. Professor Bill Wulf from the University of Virginia and Professor Brian Randell from the University of Newcastle, U.K., sponsored the workshop, and guests included representatives from Microsoft, IBM, and other major companies. The conclusions of these assembled experts were published in a set of papers. Figure 1 shows one set of predictions: The expected growth rate of four specific types of professional programmers

between now and the year 2010. Note that the vertical scale is logarithmic. You can categorize programmers in four areas:

- *IS department programmers.* These programmers work directly for an IS department. Their numbers are in fairly steep decline, and will probably reduce to the low hundred thousands by the year 2010, from about two million.
- *Software company programmers.* These programmers work directly for software companies that produce specialized or packaged applications. The experts expect their numbers to rise to the current level of IS department programmers by the year 2010.
- *Embedded software programmers.* This category of programmers produces software that is embedded in products such as cars, airplanes, consumer electronic products, and so on. According to the assembled experts, their numbers are increasing dramatically, and will likely rise to more than 10 million by 2010. Remarkably, the number of lines of C code embedded in consumer products doubles every year.
- *Occasional programmers.* This category covers professional office workers, such as accountants, media technicians, or middle managers who program sporadically as part of their professional activities. In other words, this category encompasses the professional end user. The number of such individuals will likely rise to more than 100 million by 2010.

The accuracy of these numerical projections is not particularly important; such projections are rarely accurate. What is more interesting are the trends that these projections imply. If we assume that our group of assembled experts is correct, the nature of software development is going to change. The trends imply that programming will become a general skill required for all professionals.

Taking the predictions as they stand, if you add together the IS department programmers and the software company programmers, you get an approximate horizontal line indicating a

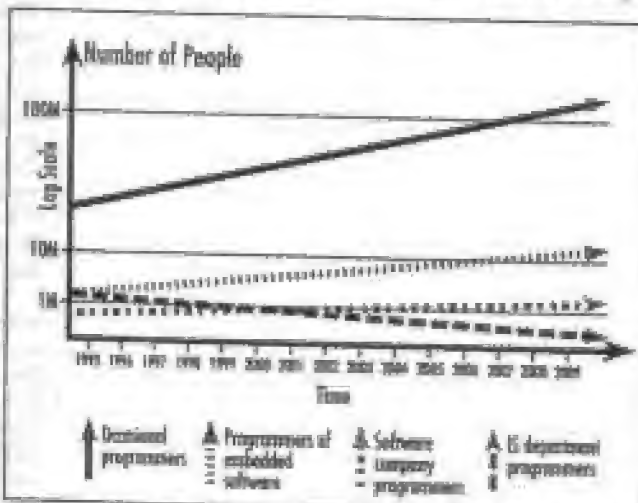


FIGURE 1 Growth-rate predictions for four categories of programmers. Notice that the vertical scale is logarithmic.

Robin Bloor is chief executive of the computer research and consulting firm ButlerBloor Ltd. You can contact his firm at 44-908-373-311 in the U.K.

constant number of two million and a few hundred thousand. In other words, as the number of IS department programmers declines, the number of software company programmers rises to compensate. However, they are the only two types of programmer that actually build corporate computer systems, whether they produce application packages or completely tailored software. Because the demand for corporate systems is unlikely to decline, these predictions suggest that either developer productivity will increase dramatically, the "occasional programmer" will fill the gap, or a combination of both.

However, these scenarios can occur only if software reuse becomes a reality. Many commentators have observed over the years that programmers are forever "reinventing the wheel"; that is, they are writing the same programs over and over again. The problem is that, so far, we have not managed to provide mechanisms that effectively enable software reusability.

Of course, many productivity tools provide reusability. Database software provides a whole host of data management capabilities that developers reuse each time they write an application. The declarative referential-integrity capability that many of the leading databases provide lets the developer specify a referential-integrity check once, which will invoke a generic routine to perform the check automatically next time. Many 4GLs provide powerful dictionary capabilities that implement reusability in a similar manner. Software tools also provide library capabilities that let developers define and reuse modules of code.

Application packages are also examples of software reusability. Any package that a multitude of companies use for the same function provides a significant level of reuse. Unfortunately, most of these packages are not easy to customize. In many instances, they provide the best economic solution, but they often involve unwelcome compromises that force a company to organize itself around the way the package works, rather than

letting it tailor the system completely to its own needs. The problem here is the level of flexibility.

Component Software

What is missing from the picture is a practical, component-based approach to building systems. This is what object orientation promises to deliver over the next decade. From the programmer's point of view, object orientation may be about inheritance, encapsulation, polymorphism, and other such things, but from the system builder's point of view, it is about tailorable and reusable software components.

Of course, the problem of providing reusable software components is not easy to resolve. The object request broker (ORB) standards promoted by the Object Management Group (OMG) are the industry's latest initiative to solve the problem. An ORB is simply a piece of software that lets software components make calls to each other and use each other's capabilities, on both the same machine and over a network. ORBs provide a more universal and capable interface than SQL can ever hope to provide.

We are now seeing the emergence of software architectures and products from key vendors that provide this capability. IBM has the System Object Model (SOM) and OSOM (Distributed SOM). HP provides the Distributed Object Management Facility. Sun provides Distributed Objects Everywhere, and Microsoft and Digital have their Common Object Model. The next thing about the ORB idea is that it caters to existing software as well as software written with OO programming languages. The idea is that you can "wrap" an old application up and treat it as a component, whether it is a single program or an entire application.

If OMG's initiative is successful, the component-based software development environment may well develop. When we have a common, low-level interface to which the industry adheres, it will be possible to write software at a component level. In essence, this is the real problem: Developers rarely design systems as a set of interacting components because they don't need to. Therefore, they end up with inflexible software packages and applications. Virtually all word processors, for example, are horrifically over-featured. They have an extraordinary level of capability that the vast majority of users doesn't even know about, never mind use. However, you cannot turn these features off, and it is not easy to integrate a word processor with another application seamlessly. Developers did not build word processors as a set of completely tailorable components. The same goes for most application packages.

The Future Programmer

The only way to make sense of the graph in Figure 1 is to assume that software development will become a matter of assembling components. If so, the graph's trend lines make sense. First of all, the number of programmers in IS departments will shrink because the amount of original development done within the IS department will diminish significantly. Most development will move out to user departments that may not understand the intricacies of large systems, but are quite capable of assembling and tailoring "business components" to carry out business tasks. On the other hand, the number of developers working for software companies will necessarily increase because they will write and maintain the majority of application components. Systems development will become more a matter of systems integration and quality control, and less a matter of producing code.

It is quite clear from the graph that the end-user community is truly in its ascendancy. The desktop computer has now become an indispensable part of the office. Pretty soon it will be the medium for all data access and communication. Consequently, programming will become a fundamental skill, but in a much simplified form. Whether the estimate of more than 100 million end-user programmers by the year 2010 is realistic is another question. We'll have to wait and see. ■

Spinning the Web

How to Provide Information on the Internet


Andrew Ford



INTERNATIONAL THOMSON PUBLISHING

I[®]T[®]P An International Thomson Publishing Company

London • New York • Bonn • Boston • Madrid • Melbourne • Mexico City • Paris • Singapore
Tokyo • Toronto • Albany, NY • Belmont, CA • Cincinnati, OH • Detroit, MI

WHARTON REPROGRAPHICS 

Spinning the Web
How to Provide Information on the Internet

International Thomson Publishing

Commissioning Editor: Liz Israel Oppedijk

Editorial Assistant: Jonathan Simpson

Van Nostrand Reinhold

Sponsoring Editor: Neil Levine

Copyright © 1995 International Thomson Publishing



A division of International Thomson Publishing Inc.
The ITP logo is a trademark under licence

All rights reserved. No part of this work covered by the copyright hereon may be reproduced or used in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems – without the written permission of the publisher.

Products and services that are referred to in this book may be either trademarks and/or registered trademarks of their respective owners. The Publisher(s) and Author(s) make no claim to these trademarks.

Made in Logotechnics C.P.C. Ltd., Sheffield

Project Management: Sandra M. Potestà

Production: Hans-Dieter Rauschner + Team

Artistic Direction: Stefano E. Potestà

Cover Illustration: William Smith

Typeset by the Author using L^AT_EX

Printed in the U.K. by Logotechnics C.P.C. Ltd., Sheffield

First printed 1995

International Thomson Publishing

Berkshire House

168–173 High Holborn

London WC1V 7AA

Van Nostrand Reinhold

115 Fifth Avenue, 4th Floor

New York, NY 10003

ISBN (ITP UK) 1-850-32141-8

ISBN (Van Nostrand Reinhold) 0-442-01996-3

1 2 3 4 5 6 7 8 9 10 EDWLT 01 00 99 98 97 96 95 94

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

Contents

Preface	xv
1 Introduction	1
1.1 What is the Web?	2
1.2 What the Web has to offer	3
1.2.1 Easy access to a global information service	3
1.2.2 Access to a global community	4
1.2.3 Front-end to existing systems	4
1.2.4 Commercial access to a global market	4
1.3 History and evolution of the Web	4
1.3.1 Background and historical context	5
1.3.2 Origins of the Web	5
1.3.3 Current mechanisms for the Web's development	6
1.4 The future of the Web	6
1.5 The organization of this book	7
I Managing a Web Site	9
2 Managing a Web Site	11
2.1 The broader issues	12
2.1.1 Connectivity	13
2.1.2 Planning for the future	13
2.1.3 Publication policy	14
2.1.4 Advertising your information	14
2.1.5 Impact on the organization	14

HTML 2.0 Quick Reference Guide

Spinning the Web

Andrew Ford

Element names are not case sensitive

Proposed features are marked thus†

Documents start with DOCTYPE followed by head and body enclosed in:

<HTML>...</HTML>

Head is enclosed in: <HEAD>...</HEAD>

Body is enclosed in: <BODY>...</BODY>

Comments are written: <!-- A Comment -->

DOCTYPE, <HTML>, <HEAD> and <BODY> may be omitted

Sample Document

```
<!DOCTYPE HTML PUBLIC "-//W3O//DTD HTML//EN//2.0">
<HTML>
  <HEAD>  <!-- A Sample Document -->
    <TITLE>Document Title</TITLE>
  </HEAD>
  <BODY>
    <H1>First Header</H1>
    <P>Paragraph one.
    <DL>
      <DT>Term<DD>Definition
    </DL>
  </BODY>
</HTML>
```

Head Elements

<TITLE>...</TITLE>

<ISINDEX>

<BASE HREF="url">

<LINK ...>

<META ...>†

<NEXTID N=id>

title (length < 64 chars)

document is searchable

base URL of document

relationships to other objects

embed meta-information, attributes:

HTTP-EQUIV, NAME, CONTENT

next identifier to be generated

Headings

Headings, level 1 to 6, specified by:

<Hn>heading text</Hn>

Spacing

<P>text...</P>

<HR>

start new paragraph

force line break

horizontal rule

Images

(level 1)

External graphics are specified with links. Embedded images are specified with:

```
<IMG SRC="url" [ALT="description"] [ALIGN="..."] [ISMAP]>
```

ALIGN can be one of: top, middle or bottom

An image map is an embedded ISMAP image that is also an anchor.

Example

```
<A HREF="http://cgi-bin/imgshow/beowulf/f196">
<IMG SRC="http://beowulf/f196.gif" ISMAP
ALT="Folio 196 of the Beowulf manuscript"></A>
```

Forms

(level 2)

Data input forms are enclosed within

```
<FORM [ACTION="..."] METHOD="..." ENCTYPE="...">...</FORM>
```

Fields defined by <TEXTAREA>, <INPUT> and <SELECT>

```
<TEXTAREA NAME="..." ROWS=r COLS=c>...</TEXTAREA>
```

multi-line text field

```
<INPUT [...]>...
```

attributes: ALIGN, CHECKED, MAXLENGTH, NAME, SIZE, SRC, TYPE and VALUE

TYPE can be one of:

checkbox, hidden, image, radio, reset, submit or text (default)

```
<SELECT NAME="..." [MULTIPLE]>...</SELECT>
```

alternatives specified by <OPTION> tags

```
<OPTION VALUE="..." [SELECTED] [DISABLED]>
```

Example

```
<HR><FORM ACTION="http://cgi-bin/script">
<P>Name: <INPUT NAME="name" TYPE="TEXT" SIZE=20>
<P>Operating System<SELECT NAME="os">
<OPTION>UNIX<OPTION>VMS<OPTION>MS-DOS</SELECT>
<P>Comment: <TEXTAREA NAME="rem" ROWS=3 COLS=20>
I think that ...</TEXTAREA>
</FORM><HR>
```

Special Characters

<	<	less than symbol
>	>	greater than symbol
&	&	ampersand
"	"	double quote
&nb		non-breaking space†

Æ	Æ	Æ	AE diphthong
Á	Á	Á	A, acute
Â	Â	Â	A, circumflex
À	À	À	A, grave
Å	Å	Å	A, ring
Ã	Ã	Ã	A, tilde
Ä	Ä	Ä	A, diæresis/umlaut
Ç	&Coedil;	Ç	C, cedilla
Ð	Ð	Ð	Eth (Icelandic)
É	É	É	E, acute
Ê	Ê	Ê	E, circumflex
È	È	È	E, grave
Ë	Ë	Ë	E, diæresis/umlaut
Í	Í	Í	I, acute
Î	Î	Î	I, circumflex
Ì	Ì	Ï	I, grave
Ï	Ï	Ï	I, diæresis/umlaut
Ñ	Ñ	Ñ	N, tilde
Ó	Ó	Ó	O, acute
Ô	Ô	Ô	O, circumflex
Ò	Ò	Ö	O, grave
Ø	Ø	Ø	O, slash
Õ	Õ	Õ	O, tilde
Ö	Ö	Ö	O, diæresis/umlaut
Þ	Þ	Þ	Thorn (Icelandic)
Ú	Ú	Ú	U, acute
Û	Û	Û	U, circumflex
Ù	Ù	Ù	U, grave
Ü	Ü	Ü	U, diæresis/umlaut
Ý	Ý	Ý	Y, acute
ß	ß	ß	German sharp s
æ	æ	æ	ae diphthong
á	á	á	a, acute
â	â	â	a, circumflex
à	à	à	a, grave
å	å	å	a, ring
ã	ã	ã	a, tilde
ä	ä	ä	a, diæresis/umlaut
ç	ç	ç	c, cedilla
ð	ð	ð	eth (Icelandic)
é	é	é	e, acute
ê	ê	ê	e, circumflex
è	è	è	e, grave
ë	ë	ë	e, diæresis/umlaut
í	í	í	i, acute
î	î	î	i, circumflex
ì	ì	ï	i, grave
ï	ï	ï	i, diæresis/umlaut
ñ	ñ	ñ	n, tilde
ó	ó	ó	o, acute
ô	ô	ô	o, circumflex
ò	ò	ò	o, grave
ø	ø	ø	o, slash
õ	õ	õ	o, tilde
ö	ö	ö	o, diæresis/umlaut
þ	þ	þ	thorn (Icelandic)
ú	ú	ú	u, acute
û	û	û	u, circumflex
ù	ù	ù	u, grave
ü	ü	ü	u, diæresis/umlaut
ý	ý	ý	y, acute
ÿ	ÿ	ÿ	y, diæresis/umlaut

Spinning the Web

How to Provide Information on the Internet

Lists

List items are preceded by , except in definition lists, which use <DT> and <DD> pairs.

...	ordered list, items numbered consecutively
...	unordered list, items marked with bullets, etc
<DIR>...</DIR>	directory list
<MENU>...</MENU>	list of short items
<DL [COMPACT]>...</DL>	definition list

Block Formatting Elements

<ADDRESS>text...</ADDRESS>	address information
<BLOCKQUOTE>text...</BLOCKQUOTE>	quoted text
<PRE [WIDTH=n]>text...</PRE>	preformatted text

Highlighting

Logical Markup

<CITE>...</CITE>	citation
<CODE>...</CODE>	code example
<DFN>...</DFN>	defining instance [†]
...	emphasis
<KBD>...</KBD>	keyboard input
<SAMP>...</SAMP>	literal characters
<STRIKE>...</STRIKE>	strike-out [†]
...	strong emphasis
<VAR>...</VAR>	variable name

Optical Markup

...	bold
<I>...</I>	italic
<TT>...</TT>	fixed-width
<U>...</U>	<u>underlined</u> [†]

Links

Anchors can be a link to another location:

```
<A HREF="url" ...>anchor text...</A>
```

or the destination for a link:

```
<A NAME="name" ...>anchor text...</A>
```

Other attributes:

4

The Hypertext Markup Language

This chapter introduces HTML (Hypertext Markup Language), the system that the Web uses for marking up documents. Later chapters look at the more advanced features such as including images in documents and setting up fill-out forms.

4.1 Overview of HTML

Web documents are written using the Hypertext Markup Language (HTML). Originally there was no rigid definition of HTML and no mechanism for extending the language. This led to the situation where different groups added features that would work with their browsers but not necessarily with other browsers. (A notable example is that of *fill-out forms*; these were added by the NCSA for the X Windows version of Mosaic, but were not supported by other browsers – not even by Mosaic on other platforms.)

Recently there has been a move to standardize HTML. The original language has retrospectively been designated version 1.0 and version 2.0 will define

30 Spinning the Web

current practice. Within version 2.0 there are three levels of features, 0, 1 and 2, which correspond to differing degrees of browser sophistication, level 0 being the simplest and level 2 the most complex. Version 2.0 is currently described in a draft Internet Request For Comments (RFC), which is expected to be finalized at the end of 1994.

HTML is a document type described in the Standard Generalized Markup Language (SGML), a system for formalizing the structure of documents and enabling documents to be interchanged between different document processing packages. Starting with version 2.0 HTML is formally defined as an SGML document type definition (DTD), which means that the definitive word on what constitutes legal HTML is embodied in an SGML definition. SGML is defined by an ISO standard (ISO 8879) and is described in *The SGML Handbook* [10] by Charles F. Goldfarb. *Practical SGML* [15] by Eric van Herwijnen is a good introduction to SGML.

Already there is discussion going on about version 3 of HTML, which will add new features to the language, such as tables, figures and mathematical formulae. This can probably be expected to appear some time in 1995.

4.2 Getting started quickly

It is useful to know how Web documents are structured, even though there are editors available that will let you create Web documents without such knowledge. HTML documents consist of plain text interspersed with markup directives, called tags. Tags are instructions to the browser software on how to display the text, and are represented by strings enclosed in angle brackets, for example <TITLE>. Figure 4.1 shows what the source for a simple HTML document looks like and Figure 4.2 shows what this looks like using X Mosaic.

```
<TITLE>An English country garden</TITLE>
```

```
<H1>An English country garden</H1>
```

```
The garden at Hidcote Manor could be said to combine the maximum formality
of design with the minimum formality of planting. It is devised as an
interconnected series of outdoor rooms, enclosed by walls or hedges, each
with a distinct theme, and each affording a tantalizing glimpse of the
next, just sufficient to lead you on to explore further.
```

```
<P> In places the garden opens out to frame a far-reaching view of the
surrounding Cotswold hills. Elsewhere the atmosphere is intimate, as in
the cottage garden where four rather dumpy topiary birds, cut from box
plants, face each other in a cosy circle.
```

Figure 4.1 HTML source for a simple Web document with minimal markup

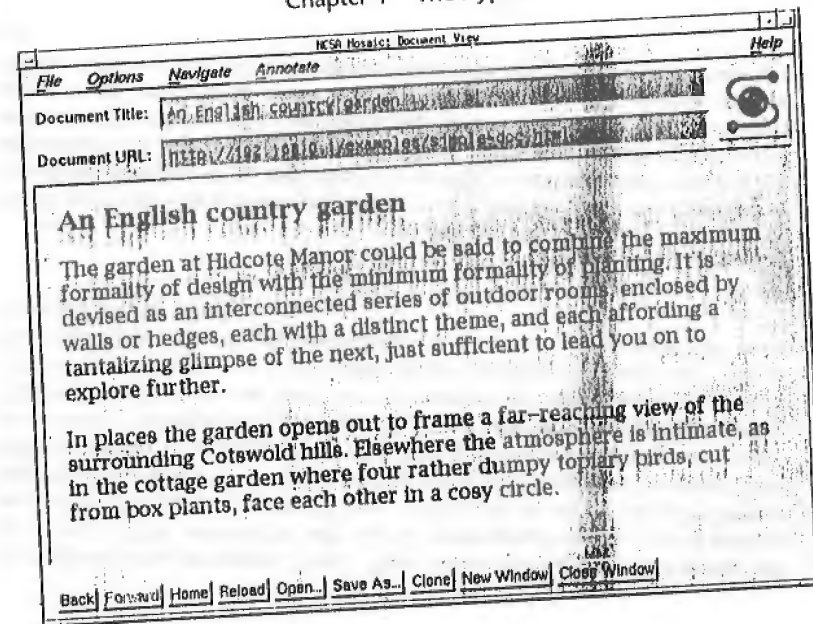


Figure 4.2 A simple Web document displayed

Tag names are not case sensitive, so <TITLE> can be written <title> or even <Title> or <Title>. Some people consider that it is easier to distinguish the tags from the text if they are written in upper case, but this is really unimportant.

To create a new paragraph in HTML you must specify a <P> tag. If you leave a blank line, browsers will ignore it. Carriage returns and blank lines are treated as a single space in HTML.

Conceptually, an HTML document consists of two parts: the head and the body. The head contains information about the document, and the body consists of the document contents. The body of the document shown in Figure 4.1 consists of a heading (contained between the <H1> and </H1> tags) followed by two paragraphs. The <P> tag for the first paragraph is omitted as its presence can be implied.

You can convert existing documents into HTML in the following way:

- If the document is stored on a word processor, save it as a plain text file
- At the top of the document add the lines:

```
<HTML>
```

32 Spinning the Web

```
<HEAD>
  <TITLE>Document Title</TITLE>
</HEAD>
<BODY>
```

replacing the literal string *Document Title* between the `<TITLE>` and `</TITLE>` tags with your document's title.

- Put a `</BODY>` tag after the last line of the text.
- Find each heading in the text and put a start heading tag at the beginning of the line and an end heading tag at the end of the line. There are six levels of heading, from `<H1>` to `<H6>`, where `<H1>` is the highest level.
- Put a `<P>` tag at the start of each paragraph in the text.

This will leave you with a document with headings and broken into paragraphs. You may want to add other features, such as emphasis or links, described later in this chapter.

A fast way to learn HTML is to look at the source of existing Web documents, particularly those you consider well put together. Most browsers have an option, *View Source*, which will pop up a window containing the raw HTML.

4.3 Structure of documents

The simple HTML document introduced in the previous section omits a number of HTML tags. If the document was prepared with an authoring tool, the missing tags would probably be supplied automatically and the source would look like Figure 4.3.

The first line is a DOCTYPE directive and says that this document uses version 2.0 HTML. If this line is omitted, version 2.0 HTML is assumed. The rest of the document is enclosed in an HTML *container element* (see below); again, this is assumed if it is omitted and is therefore not strictly necessary. Most existing browsers allow you to omit these lines. They are shown here for the sake of completeness.

HTML (and SGML) regard a document as a logical hierarchy of elements. Hence, elements are the structural components of a document. Elements start with a tag identifying their type. An element can be a single entity, such as an included image or a special character. These do not require an end tag. Alternatively an element might be a chunk of data or text, which logically requires a terminating tag, in which case the element is referred to as a container.

End tags may be omitted if the end tag can be implied by what follows. For example, `</P>` tags need not occur in a document since they are implied by a `<P>` tag or in fact by any text; similarly the first `<P>` tag after a heading can

```
<!DOCTYPE HTML PUBLIC "-//W3O//DTD WWW HTML 2.0//EN">
<HTML>
  <HEAD>
    <TITLE>An English country garden</TITLE>
  </HEAD>
  <BODY>
    <H1>An English country garden</H1>

    <P> The garden at Hidcote Manor could be said to combine the
    maximum formality of design with the minimum formality of
    planting. It is devised as an interconnected series of outdoor
    rooms, enclosed by walls or hedges, each with a distinct theme,
    and each affording a tantalizing glimpse of the next, just
    sufficient to lead you on to explore further.

    <P> In places the garden opens out to frame a far-reaching view
    of the surrounding Cotswold hills. Elsewhere the atmosphere is
    intimate, as in the cottage garden where four rather dumpy
    topiary birds, cut from box plants, face each other in a cosy
    circle.
  </BODY>
</HTML>
```

Figure 4.3 Complete markup for a simple document

be implied by the presence of text. This means that existing documents will conform to HTML version 2.0 with regard to paragraph marks, even if, looking at the raw HTML, the paragraph tags seem to be strangely placed (often at the end of the last line of the preceding paragraph from the viewpoint of an HTML version 2.0 browser). Originally `<P>` tags were paragraph separators and terminated a paragraph. They are now being redefined to start paragraphs. This is being done so that attribute information for a paragraph, such as the type of justification, can be included in the leading paragraph tag. This is also how SGML does things, and efforts are under way to make HTML more compatible with SGML.

Some start tags may contain *attributes*, which further define the characteristics of the element. Most attributes are specific to each element type and are described with their related elements below. Attributes usually consist of an attribute name followed by an equals sign and a value. The value may be a string literal enclosed in either single or double quotes or it may be a name token. It is best always to put quote characters around attribute values in a tag.

4.4 Naming schemes on the Web

The World Wide Web uses a universal naming scheme, the Uniform Resource Identifier (URI), to identify and address documents and other resources on the Net. This scheme is described in an Internet Request For Comments (RFC 1630), and encompasses a number of schemes already in general use and some which are still being developed. Two new schemes, the Uniform Resource Name (URN) and the Uniform Resource Citation (URC), are under discussion, which together will allow copies of resources to be distributed across the Web and facilitate retrieval of the closest or cheapest copy. These make use of the current scheme used on the Web, the Uniform Resource Locator (URL), which expresses the address of a resource and the method by which it can be accessed.

4.4.1 Syntax of URLs

This section describes the syntax of URLs in detail. You may want to skip straight to Section 4.5 on an initial reading.

The general syntax of a URL is:

scheme:path

The *scheme* identifies the protocol, such as HTTP, Gopher, FTP, and so on, that the browser should use to access the resource. The interpretation of *path* depends on the protocol being used. For many protocols *path* is taken to be a hierarchical name including a host name and optional port number. Host names are preceded by a double slash (//). Case may or may not be significant within the *path*, depending on the operating system on which the server is running. Port numbers are numeric identifiers that specify which server program on the server machine is addressed. These are standardized for standard protocols: Gopher uses port 70, HTTP uses 80, and so on, and where the standard port is used it need not be explicitly stated in the URL.

The *path* may be followed by a query string or a fragment identifier.

Query strings can be phrases used to locate indexed documents. They are also sometimes used to pass coordinate data from image maps and user input from forms to a server. They are indicated by a question mark (?) following the *path*. Within a query string spaces may be replaced by plus signs (+), which means that real plus signs must be encoded.

Fragment identifiers are indicated by a hash sign (#) followed by a name at the end of the URL. They are interpreted by the browser as the address of locations within a resource, and are not actually passed to the server.

Partial URLs can occur in documents. These are interpreted by the browser as being relative to the URL of the current document, using rules similar to

those used to resolve filenames on the UNIX system. The strings .. and . are taken to mean the next level up and the current level respectively.

URL-encoding

There are a number of special characters that cannot be directly included in the *path* part of a URL: +, <, >, %, ", /, ? and the space character. These can be encoded as a per cent symbol (%) followed by the hexadecimal value of the character in the ISO-8859 character set. Spaces that represent word boundaries are encoded as plus signs.

4.4.2 URLs for different information systems

This section describes the specific syntax for each of the commonly used information system protocols. The type of a document is indicated separately from the protocol. Documents of a particular type are not restricted to being retrieved using a particular protocol.

HTTP

The Hypertext Transfer Protocol is the native method of transferring documents on the Web. It is the protocol most often used for accessing HTML documents. An HTTP URL has the syntax:

`http://host[:port]/path`

The standard port for HTTP servers is port 80; if the server is listening on this port you do not have to specify the port number explicitly. Examples of HTTP URLs are:

`http://info.cern.ch/hypertext/WWW/Tools/Overview.html`

`http://wintermute.ncsa.uiuc.edu:8080/auth-tutorial/`

Gopher

Gopher is a precursor to the Web that views information as a hierarchy of menus, which may contain text and other format files. It is still in widespread use. Gopher items can be specified as:

`gopher://host[:port]/[type[item]]`

The standard port for Gopher servers is 70 and if this is used then it does not have to be specified. Gopher uses the concept of a *selector* to refer to the type of a resource and its pathname. The *type* value is encoded as a single character, the most common being 0 for text files and 1 for menus. The type

36 Spinning the Web

is explicitly included both as the first character of the pathname, or *item*, and as a separate field, and thus occurs twice after the host name.

An example of a Gopher URL is:

```
gopher://gopher.micro.umn.edu/11/
```

This refers to the top-level Gopher menu at `gopher.micro.umn.edu`, and could be abbreviated by dropping the trailing `11/`.

File Transfer Protocol

File Transfer Protocol (FTP) is one of the oldest mechanisms on the Internet for retrieving files from remote machines. Files and directory listings can be specified in URLs using the `ftp` scheme:

```
ftp://[username[:password]@]host/path/file
```

By default the username is *anonymous*, the username for anonymous file transfer. Explicitly specifying a different username is not recommended if a password is required for that account as it will have to be encoded in the URL, which may be stored in plain text in documents.

An example of an FTP URL is:

```
ftp://ftp.w3.org/pub/ls-LR.Z
```

File

The `file` scheme allows files on your local system to be specified, in which case the browser will read the file directly. The syntax is:

```
file://hostname/path
```

A host name can be specified so that browsers can determine that the file referred to is not on the local system. They may then use another scheme, such as *anonymous FTP*, to try to retrieve the file.

An example of a file URL is:

```
file:/usr/local/lib/ghostscript/README
```

News

The `news` scheme allows USENET news groups or articles to be specified. It differs from the other schemes in that no host is specified in the URL; your news host is usually specified directly to the browser by an environment variable or some other means.

The syntax for newsgroups is:

```
news:newsgroup
```

and for individual articles:

```
news:article-id
```

Examples of actual news URLs are:

```
news:comp.infosystems.www.providers
```

```
news:bwh.2.0010809C@access.digex.net
```

4.5 Basic HTML elements in detail

This section describes the basic HTML elements.

4.5.1 Comments

Comments can appear anywhere in an HTML document, except within a tag, and are enclosed between the strings `<!--` and `-->`, like this:

```
<!-- This is a comment -->
```

Comments may not be nested: browsers will regard everything after the first comment terminator as markup.

Although strictly speaking the end of comment is denoted by the string `-->`, some browsers regard the single character `>`, without the preceding double hyphen, as a comment terminator.

4.5.2 The DOCTYPE directive

The `DOCTYPE` directive is an SGML construct that identifies the type of the document as being HTML. This should be:

```
<!DOCTYPE HTML PUBLIC "-//W3O//DTD W3 HTML 2.0//EN">
```

If the `DOCTYPE` directive is missing, this information should be assumed by browsers compliant with HTML version 2.0 or later.

4.5.3 The document head

The head contains meta-information (information of a higher order) about the document, such as the title. The head is identified by the `HEAD` element. This can be omitted, but it is better to include it as it allows server software to find out information about the document without having to search through the whole document.

The following elements can occur in the head:

TITLE	The title of the document.
ISINDEX	Indicates that the document is searchable.
BASE	Specifies the URL of the document.
LINK	Specifies relationships to other documents.
NEXTID	Indicates the next identifier to be generated (for use by authoring tools).
META	Specifies meta-information about the document.

None of the head elements are compulsory, although a TITLE element is recommended.

The TITLE element probably needs no further explanation, but note that only ASCII characters may be included and that the length should not exceed 64 characters including spaces.

The ISINDEX element tells the browser that the document can have a *query string* appended to its URL and the server will then invoke a script to perform a search accordingly. This is described in more detail in Chapter 11.

The BASE element takes a single attribute, HREF, which gives the URL of the document.

The LINK element describes the relationship of the document to other documents.

The NEXTID element specifies the next anchor label to be automatically generated within the document. This is used by HTML editing tools, to keep track of hypertext link labels (see Section 4.6). It has no meaning to a browser and you don't need to use it if you are writing a document by hand.

The META element was introduced in HTML version 2.0 as a 'catch-all' to allow meta-information that isn't covered by any of the other head elements to be included. It takes three attributes: NAME, HTTP_EQUIV and CONTENT. The information is named either by the NAME or HTTP_EQUIV element. For example:

```
<META HTTP-EQUIV="Expires"
  CONTENT="Tuesday, 19-Apr-94 18:47:05 GMT">
```

4.5.4 The document body

The body of a document comprises the actual document contents to be displayed. This includes headings, text and images.

Headings

You can have up to six levels of heading in your documents, marked from the highest level, H1, to the lowest, H6; for example, the top-level heading:

```
<H1>An English Country Garden</H1>
```

may be followed at the next level by:

```
<H2>The Vegetable Plot</H2>
```

Paragraphs and line breaks

Unlike other document markup systems, HTML ignores empty lines embedded in the document source and runs the text together into a single paragraph on the screen. Paragraph breaks must be explicitly marked with the <P> tag. The </P> end tag can be omitted, but you may find that these are automatically inserted by HTML authoring tools.

The <P> tag usually generates extra vertical space between paragraphs. If you want to start a new line without extra vertical space, use the
 tag. This is an empty element, which means it does not have an end tag. The
 tag is often used within the ADDRESS element (discussed later in this chapter) to separate lines of an address.

The <HR> tag, which is also an empty element, creates a horizontal dividing line across the screen. It is often used to separate blocks of information or to visually delineate fill-out forms.

Special characters

HTML uses the character < to start a tag, so you cannot use this character without a browser interpreting it as markup. Similarly the double quote character is used to start and end attribute value strings.

In order to represent these characters in your HTML documents you must use the entities < and ". To get a literal & you must use the entity &. Although HTML uses ISO 8859 for its character set, entities can also be used for non-ASCII characters such as accented characters in case you cannot enter such characters directly from the keyboard. Table 4.1 contains a list of the standard entities.

List elements

There are a number of HTML elements for defining different types of list within the document body:

- Unordered lists (UL)

Table 4.1 ISO Latin 1 Entities in HTML

Á	â	À	A, acute accent	á	â	á	a, acute accent
Â	â	Á	A, circumflex accent	â	â	â	â, circumflex accent
Ã	à	À	A, grave accent	ã	à	ã	ã, grave accent
Ä	å	Ä	A, ring	ä	å	ä	ä, ring
Å	å	Å	A, tilde	å	å	å	å, tilde
Æ	å	Ä	A, dieresis/umlaut	æ	å	ä	ä, dieresis/umlaut
Ç	å	Å	A, cedilla	ç	å	å	å, cedilla
È	ð	Ê	E, (Icelandic)	è	ð	ê	ê, (Icelandic)
É	ð	É	E, acute accent	é	ð	é	é, acute accent
Ê	ð	Ê	E, circumflex accent	ê	ð	ê	ê, circumflex accent
Ë	ð	Ë	E, grave accent	ë	ð	ë	ë, grave accent
Ì	ð	Ê	E, dieresis/umlaut	ì	ð	ê	ê, dieresis/umlaut
Í	ð	Ë	E, acute accent	í	ð	ë	ë, acute accent
Î	ð	Ê	E, circumflex accent	î	ð	ê	ê, circumflex accent
Ï	ð	Ë	E, grave accent	ï	ð	ë	ë, grave accent
Ð	ð	Ê	E, dieresis/umlaut	ð	ð	ê	ê, dieresis/umlaut
Ñ	ð	Ë	E, acute accent	ñ	ð	ë	ë, acute accent
Ò	ð	Ê	E, circumflex accent	ò	ð	ê	ê, circumflex accent
Ó	ð	Ë	E, grave accent	ó	ð	ë	ë, grave accent
Ô	ð	Ê	E, dieresis/umlaut	ô	ð	ê	ê, dieresis/umlaut
Õ	ð	Ë	E, acute accent	õ	ð	ë	ë, acute accent
Ö	ð	Ê	E, circumflex accent	ö	ð	ê	ê, circumflex accent
×	ð	Ë	E, grave accent	÷	ð	ë	ë, grave accent
Ø	ð	Ê	E, dieresis/umlaut	ø	ð	ê	ê, dieresis/umlaut
Ù	ð	Ë	E, acute accent	ù	ð	ë	ë, acute accent
Ú	ð	Ê	E, circumflex accent	ú	ð	ê	ê, circumflex accent
Û	ð	Ë	E, grave accent	û	ð	ë	ë, grave accent
Ü	ð	Ê	E, dieresis/umlaut	ü	ð	ê	ê, dieresis/umlaut
Ý	ð	Ë	E, acute accent	ý	ð	ë	ë, acute accent
Þ	ð	Ê	E, circumflex accent	þ	ð	ê	ê, circumflex accent
ß	ð	Ë	E, grave accent	ÿ	ð	ë	ë, grave accent

- Ordered lists (OL)
- Definition lists (DL)
- Directory lists (DIR)
- Menus (MENU).

Lists may be nested and different types of list may be nested within each other.

Unordered lists are displayed as lists of bullet items. Individual items within a list can be quite large – up to several paragraphs, and may contain elements such as images, hypertext links or other lists. Each list item is identified by an LI element. The end tag is optional and in fact some browsers do not recognize it.

```
<H2>Plant Classification</H2>
<UL>
<LI> annuals
<LI> biennials
<LI> perennials
</UL>
```



Figure 4.4 An unordered list

Ordered lists are displayed as lists of numbered items. Each list item is identified by an LI element, as with unordered lists. The items are numbered automatically by the browser and the lists can be nested. Exactly how nested numbered lists are displayed is determined by the browser.

```
<H2>The Seasons</H2>
<OL>
<LI> Spring
<LI> Summer
<LI> Autumn
<LI> Winter
</OL>
```

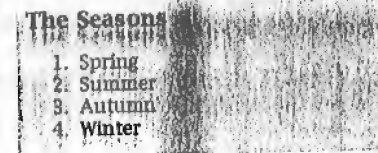


Figure 4.5 An ordered list

Both menu lists and directory lists are variants of unordered lists, and are intended for lists of short items that can be displayed in a compact style. The items on a menu list are frequently set up as hypertext links to create the functionality of a menu. Each menu list item should be a single line and a directory list item should not be longer than 20 characters. Some browsers display menu or directory lists in the same way as unordered lists, while others display them without the bullets that are characteristic of unordered lists.

42 Spinning the Web

```
<H1>Table of Contents</H1>

<MENU>
<LI> <A HREF="#section1">Section 1</A>
<LI> <A HREF="#section2">Section 2</A>
<LI> <A HREF="#section3">Section 3</A>
</MENU>
```

Definition lists are intended for lists of terms and their definitions. The term is preceded by a <DT> tag and the definition by a <DD> tag. It is permissible to have a number of terms preceding one definition. Definition lists are often used for glossaries, for example. Figure 4.6 illustrates the use of description lists and Figure 4.7 shows how it is displayed by the X Mosaic browser.

```
<TITLE>Parts of a plant</TITLE>
<H2>Parts of a plant</H2>

<DL>
<DT> Bract
<DD> Leaf below the <EM>calyx</EM>.
<DT> Calyx
<DD> Circle of leaf-like material which forms the outer case of
a flower bud.
<DT> Petiole
<DD> The stalk joining a leaf to a stem.
<DT> Spadix
<DD> Closely arranged spike of flowers, usually enclosed by a
<EM>spathe</EM>.
<DT> Spathe
<DD> Large <EM>bract</EM> or pair of bracts enclosing the
<EM>spadix</EM>.
</DL>
```

Figure 4.6 HTML for a sample description list

The COMPACT attribute can be specified in the <DL> tag to suggest that the browser should display the definition list in a compact form, minimizing the amount of space between successive pairs of items. It may also reduce the width of the term (DT) column.

Definition lists can be used to create fancy bullet lists using an icon in each DT element, as shown in Figures 4.8 and 4.9. Purists consider this to be a misuse of the construct, but currently there is no other way to achieve this effect within HTML.

Highlighting

HTML has a number of elements for highlighting text, which can be categorized as logical markup and visual markup elements. Some of the highlighting

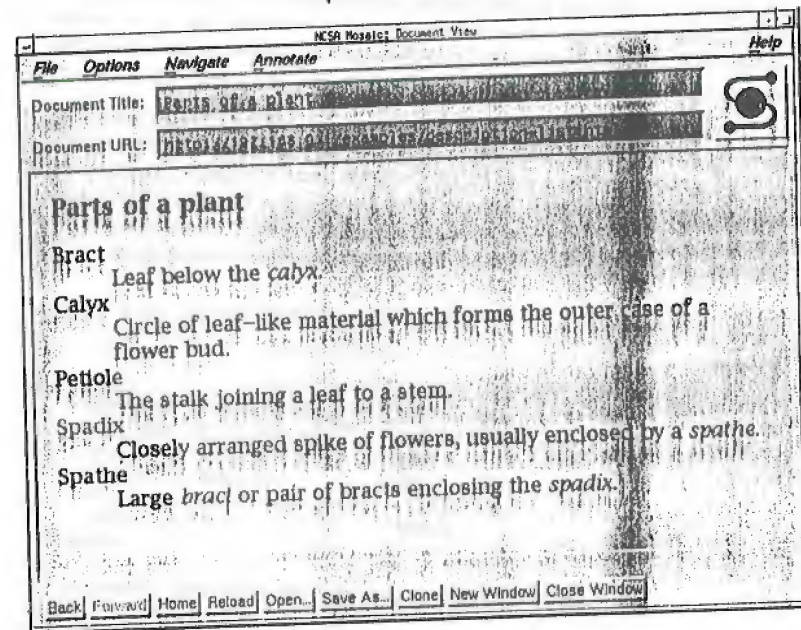


Figure 4.7 A sample description list

elements have not yet been ratified. These elements are indicated with the word *proposed* in the lists below.

The following elements are logical markup elements:

<CITE>...</CITE>	citation
<CODE>...</CODE>	code
<DFN>...</DFN>	defining instance (proposed)
...	Emphasised Text
<KBD>...</KBD>	Keyboard
<SAMP>...</SAMP>	literal characters
<STRIKE>...</STRIKE>	struck-out text (proposed)
...	strong emphasis
<VAR>...</VAR>	variable name

The following elements are visual markup elements:

...	bold
<I>...</I>	italic
<TT>...</TT>	fixed-width
<U>...</U>	underlined (proposed)

Logical markup is generally preferred to visual markup as the interpretation


```

<TITLE>Trees and Shrubs</TITLE>
<H2>Trees and Shrubs</H2>

<DL COMPACT>
<DT> <IMG SRC="images/deciduous.xbm">
<DD> Quercus robur: the common oak,
large deciduous tree.

<DT> <IMG SRC="images/conifer.xbm">
<DD> Metasequoia glyptostroboides:
the dawn redwood, vigorous conifer.

<DT> <IMG SRC="images/evergreen-shrub.xbm">
<DD> Choisya ternata: the Mexican orange
blossom, evergreen shrub.

<DT> <IMG SRC="images/decid-shrub.xbm">
<DD> Philadelphus delavayi: mock orange,
large deciduous shrub.
</DL>

```

Figure 4.8 Using icons in a definition list

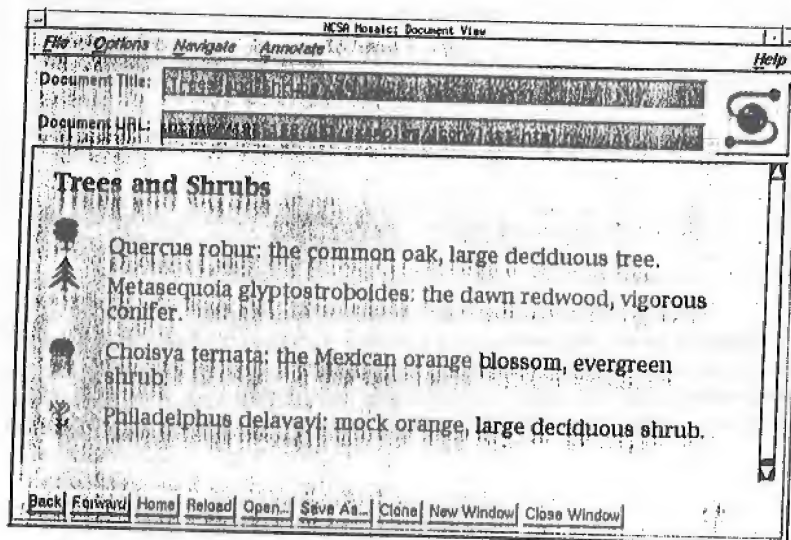


Figure 4.9 Using icons in a list

is less rigid and more configurable.

Block formatting elements

There are three block formatting elements: ADDRESS, BLOCKQUOTE and PRE.

The ADDRESS element is used to format postal addresses, signatures, email addresses and information of this type. The content is generally displayed in an italic font, indented or right justified. This element is often added at the bottom of a document giving the author and date that the document was last changed, for example:

```

<ADDRESS>
  Andrew Ford (A.Ford@icarus.demon.co.uk), 28 October 1994
</ADDRESS>

```

The BLOCKQUOTE element is used for including quotations in a document. A new paragraph is started and text is indented both left and right. The browser may display this text in a different font.

```

<TITLE>Example of preformatted text</TITLE>
<H1>Seed Sowing Times in the United Kingdom</H1>

<PRE>
Name                               Month
-----
Acanthus mollis                   Apr - May
Dianthus neglectus                Apr
Helleborus orientalis            May - Jun
Papaver somniferum               Apr - May
</PRE>

```

Figure 4.10 HTML for preformatted text

Sections of preformatted text can be included in HTML documents by using the PRE element. These are displayed in a fixed-width font and can be useful when the formatting of your information is critical but HTML does not provide the facilities to format your information as you want. An example is tabular material (until HTML version 3.0 is in widespread use). This is illustrated in Figures 4.10 and 4.11.

Between the <PRE> and </PRE> tags line boundaries are respected. Anchors and character formatting are allowed and tab characters expand to one or more spaces such that the next character appears on a character position that is a multiple of eight. All text formatted using the PRE element will be displayed by



Figure 4.11 Example of preformatted text

browsers in a typewriter font. The `<PRE>` tag can take one optional attribute, `WIDTH`, which defines the width of the text in characters.

When using `<PRE>` put the matching `</PRE>` at the start of a line, since an extra blank line may be inserted if there are spaces before it.

4.6 Hypertext links

A hypertext link is a pointer from a place in a document to another destination. At its simplest this destination is a different document. The destination might be a resource other than a document, such as an external image, a video clip or a sound file, or a labelled point in the original document, or a labelled point in a different document. Hypertext links are what puts the *hyper* into hypertext!

Hypertext links referring to non-HTML resources usually cause an external program, a viewer or helper application, to be invoked by the browser to display or play the resource. Setting up browsers, including the use of external viewer programs, is discussed in Section 9.4.1.

Both the starting point and the destination of a hypertext link are referred to as **anchors** and are marked by the anchor tag `<A>`. Anchors *may* have several attributes, but they *must* have one or both of the `NAME` and the `HREF` attributes.

```
<A NAME="name" HREF="dest-url">highlighted text</A>
```

The `HREF` attribute specifies that the anchor is the start of a hypertext link and the attribute value (*dest-url*) is the destination URL. The browser highlights

the text between the `<A>` and `` tags and interprets clicks on the text as a request for the document referenced. The text should follow on immediately after the `<A>` tag and the `` end tag should immediately follow the text, with no embedded spaces, otherwise space characters will be highlighted, which looks silly.

The `NAME` attribute specifies that the anchor is the destination of a link which has been set up elsewhere.

The remaining attributes, `METHODS`, `REL`, `REV` and `URN`, also optional, are not commonly used or supported by browsers. Their syntax is described in the HTML specification but their functionality is still being discussed by the Web development community.

Here are some examples of how different types of hypertext links can be set up.

```
<A HREF="http://www.somesite.org/doc.html">description</A>
```

This code could be used to create a hypertext link from one location in a document to another document. `http://www.somesite.org/document.html` is the URL of the destination document, and `description` will appear as highlighted text in the original document.

Lines referencing a non-document resource such as an image, audio or video clip are set up in a similar way:

```
<A HREF="http://www.somesite.org/image.gif">an image</A>
<A HREF="http://www.somesite.org/audio.au">a sound</A>
<A HREF="http://www.somesite.org/video.mov">a movie</A>
```

The browser makes no interpretation of the URL, merely sending a request to the named server, which will determine the type of resource, generally from the filename extension, and send the resource together with an indication of its type to the browser. The browser will attempt to start an external application to display any resource that it cannot handle itself.

Hypertext links can be created from one location in a document to another location in the same document:

```
<A HREF="#next_topic">jump to the next topic</A>
```

The destination must have a named anchor:

```
<A NAME="next_topic">text</A>
```

You can also make a link to a named anchor in a separate document:

```
<A HREF="http://www.somesite.org/doc.html#next_topic">jump
to the next topic</A>
```

48 Spinning the Web

When the anchor is selected a browser may use the optional TITLE attribute to display the title of the document being fetched. Of course, the title specified in the anchor may be different from that in the document being fetched.

```
<A HREF="http://www.somesite.org/document.html"  
  TITLE="An English Country Garden">an English garden</A>
```

Notes for:
An Introduction to Decision Technologies:
Using the Computer as an Analysis Tool¹

Steven O. Kimbrough
James D. Laing
Gerald L. Lohse
University of Pennsylvania
The Wharton School
3620 Locust Walk, Suite 1300
Philadelphia, PA 19104-6366
Fax: 215-898-3664

Draft: December 22, 1995

¹File: dopim101dtbook. Created: May 27, 1995.

Chapter 3

A Brief Introduction to Decision Analysis¹

We'll start simply, with *decision trees*. I'll then say a word about *utility theory*, then something about *multiattribute utility theory*. These build upon *decision trees*.

3.1 Decision Trees and Their Analysis

Usually, when we are faced with a decision we are also faced with some significant related *uncertainty*. We must decide for whom to vote, but we are uncertain about who would be the most effective leader. We must decide how to price a particular product, but we are unsure of how various prices will affect sales of the product. And so on. It is in fact quite easy to think up examples of decisions that need to be made in the presence of significant departures from full knowledge of the consequences. Indeed, such decision contexts are the rule, not the exception. Let's now consider a simple example, a simple decision problem that we shall draw lessons from and model.

¹File: dt-decision-analysis. Revised: 951322, 951128, 951022, 951023. From: (MISNotes-decision-analysis. Revised: 950919).

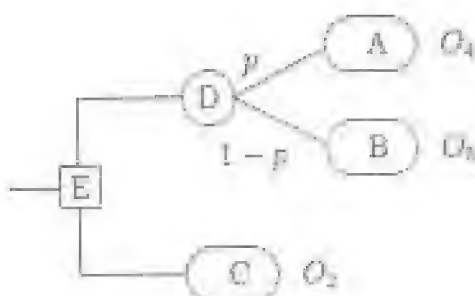


Figure 3.1: Decision Tree for the Parking Meter Problem

3.1.1 A Simple Example

Consider the parking meter problem. You have parked your car in a metered slot and the meter has run out. Your decision is whether or not to plug the meter. If you plug the meter, it will cost you, say \$1.50, but you are sure of not getting a ticket. If you do not plug the meter, then you may get a ticket (for \$15.00) or you may not, depending on whether the parking enforcement person comes by during the next hour.

Do you want the sure thing, for a cost of \$1.50, or do you want to take the chance that you will get a ticket? Suppose you could make a pretty solid guess at the probability of getting a ticket, and say that probability was 0.4. You might then reason roughly as follows. If you plug the meter, you will be out \$1.50 for sure. If you do not plug the meter you have a 60% chance of being out nothing, and a 40% chance of being out \$15.32 (the ticket plus the postage stamp to mail your money in). On average, if you do not plug the meter, you expect lose $\$0.00 \cdot 0.6 + \$15.32 \cdot 0.4 \approx \6.12 . This is considerably more than \$1.50, so you ought to plug the meter.

It is helpful, both here and in general, to draw a diagram to represent the situation. Figure 3.1 is such a diagram—called a *decision tree*—for the parking meter problem. The diagram will help us understand how to generalize the sort of reasoning just expressed. In the diagram, we begin with the

node labelled E. The fact that it is a box indicates that it is a decision node in the tree. The decision at hand is whether to plug the meter (follow the line to node C) or not to plug the meter (follow the line to node D). If we plug the meter, we arrive at a sure outcome, node C (indicated by an oval), with outcome O_2 . In our example, $O_2 = -\$1.50$.

If we do not plug the meter, we arrive at a chance node, node D (indicated by a circle), where one of two things could happen. First, we might, with probability p (equal to 0.4 in our example) get a ticket. If so, then we get to node A, an outcome node as indicated by the oval, with outcome O_4 (equal to $-\$15.32$ in our example). Second, we might not get a ticket. The probability of this is $1 - p = 1 - 0.4 = 0.6$. If so, then we arrive at outcome node B and receive $O_5 = \$0.00$ in our example.

In terms of our decision tree, the reasoning we went through above might be expressed as follows. Node C has a value, O_2 . If node D had a value, V_D , then I could make my decision (at node E) simply by choosing the larger of O_2 and V_D . I can get a reasonable value for node D (since it is a chance node) by getting the average or expected value of the node. That value is $p \cdot O_4 + (1 - p) \cdot O_5 = -0.4 \cdot \$15.32 - 0.6 \cdot \$0.00 = -\6.128 .

Thus, we have illustrated how decision trees are *folded back* or *pruned*. What has worked in this example works generally. The idea is to work backwards (conventionally, from right to left in the tree diagram) from outcome nodes to decision or chance nodes, assigning values to the nodes until all nodes are given a value. Values of the outcome nodes are assumed given. The value of a chance node is the expected, or weighted average, value of its daughter nodes (nodes, conventionally, to the right). The value of a decision node is the maximum of the values of its daughter nodes.

3.1.2 Comments

1. The "right" decision to make is the decision with the highest expected value. We assume that our decision trees begin on the left with a decision node representing our fundamental choice. Should we plug the meter or not?
2. The parking meter problem is perhaps the simplest of all nontrivial decision trees, but it is also a model for many real problems.

3. Decision trees may be elaborated to essentially an arbitrary level of complexity. Basically, this is done by any or all of the following:
 - (a) Replace an outcome node with a tree, e.g., with a chance node followed by two (or more) outcome nodes.
 - (b) Add one or more daughter nodes to a chance node.
 - (c) Add one or more daughter nodes to a decision node.

But remember: a decision tree is a model of a real situation. In building any model, judicious decisions must be made balancing fidelity to the real phenomena with simplicity and tractability. Here, as elsewhere, the KISS (= keep it simple, stupid) injunction is well worth remembering. Also: Think of the outcome nodes as representing a large amount of unmodeled information. They are stopping points, perhaps only temporarily, in the analysis. If further reflection reveals that the modeling needs refinement, then it is a simple matter to replace outcome nodes with more extensive trees.

4. As models, decision trees should always be subjected to *post-evaluation analysis*. Particularly important is *sensitivity analysis*. Do small changes in the model's parameters greatly affect the recommended decision? For the parking meter problem, the parameters are the values of the three outcomes and the probability of getting a ticket. These are all exogenous to the problem, but need to be examined.

One way of performing such sensitivity analysis is to ask a series of what-if questions. In a spreadsheet environment this can often be accomplished simply by changing the value in a cell that holds the value of the parameter in question, then having the spreadsheet recalculate. Nothing could be easier. What-if analysis is very useful in general, but limited by the fact that it is not systematic. Spreadsheet programs typically offer a one-way and two-way *data table* capability. This can be used to perform post-evaluation analysis in a more systematic fashion. Further, spreadsheets tend to offer *goal-seeking* features. These, too, can be very useful for post-evaluation analysis. For example, in the parking meter problem we might want to ask how low the probability of getting a ticket would have to go before we would decide not to plug the meter.

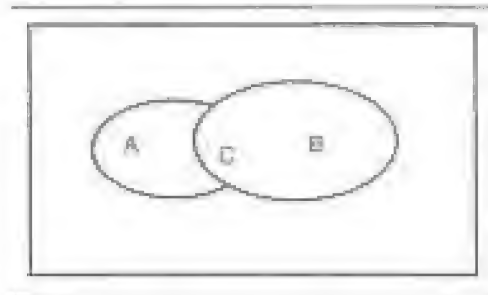


Figure 3.2: Venn Diagram of Probabilities

5. There are many other interesting and useful manipulations of decision trees. For the most we shall resist the temptation to discuss them, excepting the discussion next, in §3.3. For information on these and other decision tree manipulations, see the references mentioned in §3.7.

3.2 Conditional Probability

Before we go further, it will be useful to review some basic probability theory, especially with regard to *conditional probability*.

In general, for any two events (or sets of events) α and β , $P(\alpha|\beta)$ is the probability that α occurs given that β occurs, and it is defined as follows:

$$P(\alpha|\beta) \stackrel{\text{def}}{=} \frac{P(\alpha \cap \beta)}{P(\beta)} \quad (3.1)$$

The expression $(\alpha \cap \beta)$ means “ α and β ” and is said to represent the *intersection* of events (or sets) α and β . Similarly, the expression $(\alpha \cup \beta)$ means “ α or β ” and is said to represent the *union* of events (or sets) α and β . So, $P(\alpha \cap \beta)$ represents the probability that both events, α and β , occur. Similarly, $P(\alpha \cup \beta)$ represents the probability that either event α occurs or event β occurs, or they both occur.

Now, it is important for you convince yourself that the definition of conditional probability, equation 3.1, is in fact a good one. Let us try drawing some Venn diagrams and thinking about the definition.

In figure 3.2 we have a representation of all possible events, U , by the area within the rectangle. We assume that $P(U) = 1$, that is, the probability of any event being within U is 1. Equivalently, $P(\bar{U}) = P(U') =$

\emptyset , that is, the probability of any event being outside U is zero. For example, if we were tossing a die, then the set of possible outcomes, \mathcal{O} , is $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$. U , the set of possible events, encompasses any subset of \mathcal{O} , including the null set, \emptyset , as well as unions and intersections of events. Thus, if α and β are events, then so are their union, $(\alpha \cup \beta)$, and their intersection, $(\alpha \cap \beta)$. Finally, if α is an event, then so is $\bar{\alpha} (= \alpha')$.

Inside, in the space of possible events, we have two ovals. The left oval represents the A event, the right oval the B event, and C the event consisting of the intersection of A and B , i.e., $C = (A \cap B)$. For example, if we were tossing a die, A might stand for the event of getting an even number and B for the event of getting a number less than or equal to 3. C , the intersection, is the event of getting a 2. What is the probability of A ? It is the probability of getting a 2 or a 4 or a 6, i.e., $P(A) = P(\{2\} \cup \{4\} \cup \{6\})$. The probability of B is $P(B) = P(\{1\} \cup \{2\} \cup \{3\})$. What's the probability of A , given that B has occurred, i.e., $P(A|B)$? Reflection on the diagram, figure 3.2, would suggest that A occurs when B occurs only if C occurs and the probability of C when B occurs is just the ratio of $P(C)$ to $P(B)$. But, since $P(C) = P(A \cap B)$, this is exactly what the formula (equation 3.1), i.e., the definition of conditional probability, tells us. Indeed, the definition is a good one.

3.3 More Information

Suppose you are standing near your parking meter, having reflected on whether to plug the meter. Just as you are about to insert the coins, an enterprising street person approaches you with an offer to sell you information as to the whereabouts of the parking enforcement personnel. Your decision problem is now complicated and you need to deliberate further. There are three cases for you to consider:

1. The street person will report with completely accurate information. That is, if the street person reports that there will be no visit from a representative of the parking authorities, then there will be no such visit (and you will not get a ticket) during the next hour; and similarly otherwise; the street person is entirely reliable. In this case, you need to determine EVPI, the expected value of perfect information (since you are offered perfect information), and use the results of this calculation to assess the price being asked by the street person.

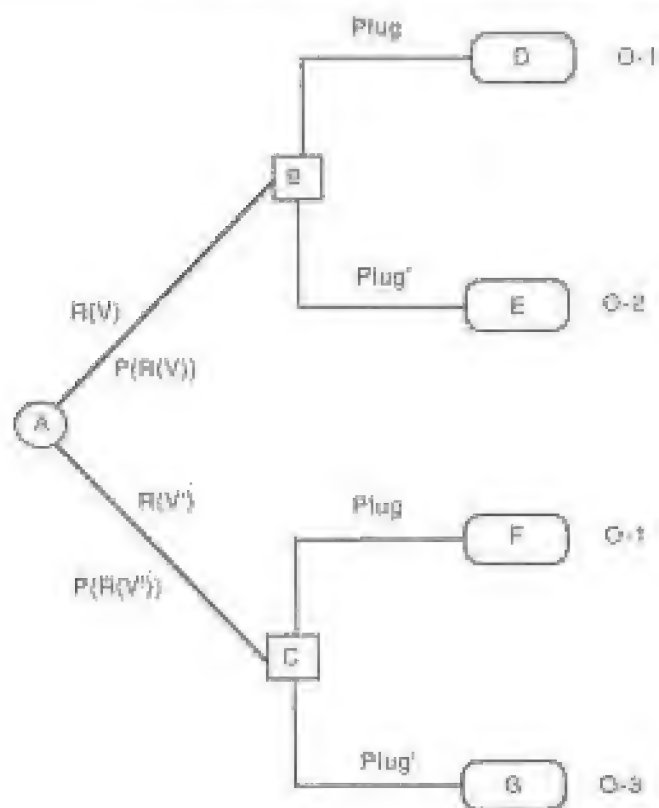
2. The street person will report with probabilistically accurate information and you know (we assume with certainty) the reliability of the street person's reports, in the sense that you know the probability of a visit by the parking authorities, given that the street person says there will be a visit, and so on. In this case, you need to determine the EVSI (expected value of sample information) with outcome-given-report information. You will use the results of these calculations to assess the price being asked by the street person.
3. Finally, the street person may report with probabilistically accurate information and you also know (we assume with certainty) the reliability of the street person's reports, in the sense that you know the probabilities of making the various reports, given that the various outcomes will occur. For example, you know the probability of the street person's saying "There will be a visit by the parking authorities during the next hour," given that there will not be such a visit. Note that this is distinctly different from the case (above) of EVSI with outcome-given-report information. There you have, e.g., the probability of there not being a visit by the authorities given that the report says "There will be a visit by the parking authorities during the next hour." We call this third case the case of EVSI with report-given-outcome information.

We now consider each of the three cases individually. In each case, we will alter our original decision tree, Figure 3.1, calculate or determine some new probabilities, and fold back the tree.

3.3.1 EVPI: Expected Value of Perfect Information

Suppose the street person's report will be entirely accurate, and we could obtain the report without cost. This results in a revised decision tree, shown in Figure 3.3.

Notice that the essential change is that the event providing information about whether there will be a visit by the parking authorities (node A in Figure 3.3) *precedes* rather than *follows* (as in Figure 3.1) any decision on whether to plug the meter. This is equivalent to having nature move "first," and nature's choice is revealed to you *before* you have to decide whether to plug the meter. This ought to be helpful, and it is. The expected value of the tree with perfect information (i.e., the tree or tree fragment in Figure 3.3) is



file: dt-parking-perfect 951023

Figure 3.3: The Parking Meter Problem with Perfect Information. Note: V' (Plug', etc.) and \bar{V} (Plug, etc.) are alternate notations for the complement of the set V (Plug, etc.).

$0.4 \cdot -\$1.50 + 0.6 \cdot \$0.00 = -\$0.60$. Why? Begin at the left of the tree in Figure 3.3. The report will either be favorable ($R(\bar{V}) = \text{"No visit"}$) or unfavorable ($R(V) = \text{"Beware, a visitor will come!"}$). Since the probability of a visit is, as before, 0.4, then we must assume that the probability that this perfectly reliable reporter will report the unfavorable outcome with probability equal, as before, to 0.4. That is, $P(V) = P(R(V))$. A similar argument applies to any other possible reports, although here there is just one: $P(\bar{V}) = P(R(\bar{V}))$. So, with probability of 0.4, you will hear "Beware, a visitor will come!" and you plug the meter for \$1.50, since $O-1 = -\$1.50$ and $O-2 = -\$15.32$. With probability of 0.6, you will hear "No visit," and you will not plug the meter, since $O-1 = -\$1.50$ and $O-3 = \$0.00$.

On average, your expected loss is $-\$0.60$. This is the expected value of your decision with perfect information. The expected value of your decision without perfect information (or any new information at all) is, as we have seen, $-\$1.50$. The EVPI for your decision is just the difference:

$$\$0.90 = (-\$0.60) - (-\$1.50) \quad (3.2)$$

Here, and in general, the expected value of perfect information for a decision is equal to the expected value of the decision with perfect information, minus the expected value of the decision without additional information. Think of it this way:

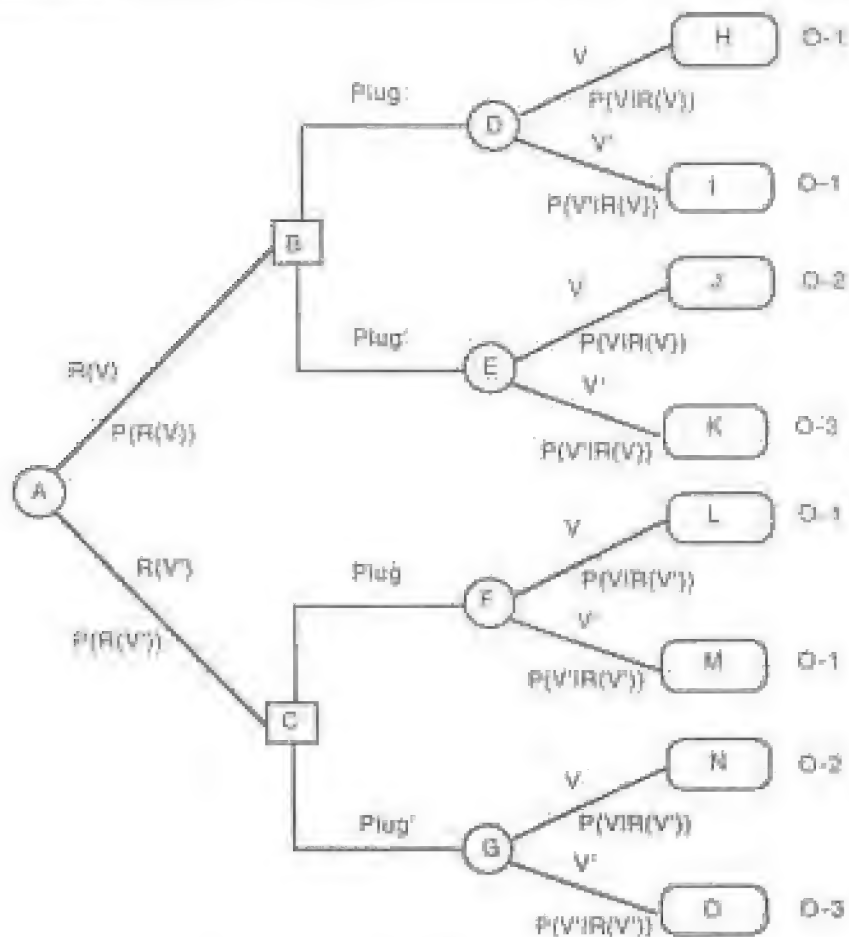
$$EVPI = EVofPI = EVDwithPI - EVDwithPriors \quad (3.3)$$

So, what do you do? Assuming you are still working under the assumption of deciding based on expected monetary value, you should only buy the information if it costs less than \$0.90.

3.3.2 EVSI with Outcome-Given-Report Information

Of course, it is unrealistic to think of the street person approaching you with a business proposition as being perfectly reliable. What this person offers may not have the value of perfect information, but it may still have value and you may rationally want to listen to it.

Figure 3.4 shows the basic scenario for the parking meter problem in the presence of imperfect information. As before $O-1 = -\$1.50$, $O-2 = -\$15.32$, and $O-3 = \$0.00$.



file: di-parking-imperfect 951023

Figure 3.4: The Parking Meter Problem with Imperfect Information

As in the case of EVPI, we begin on the left with a chance node whose daughter nodes, $R(V)$ ("Beware, a visitor will come!") and $R(\bar{V})$ ("No visit"), represent the possible outcomes of the report you are considering to buy. Represent the probabilities of these events as $P(R(V))$ and $P(R(\bar{V}))$, where $P(R(\bar{V})) = 1 - P(R(V))$. If the report is unfavorable, $R(V)$, the outcome may or may not be favorable (V , a visit or not, \bar{V}). These probabilities are now conditioned upon the outcome of the report. That is, e.g., given that the report is unfavorable, there is a probability that the actual outcome is favorable. We symbolize this probability as $P(\bar{V}|R(V))$.

The tree in Figure 3.4 has all of its chance node daughter branches labeled symbolically, with $P(R(V))$, $P(V|R(V))$, and so on. If we are to use this decision tree to make decisions, we will need to determine the actual values for these probabilities. If all the actual values are given, then we can fold back the tree to get the expected value of the decision with sample information (EVDwithSI), and we can then calculate EVSI much as we did EVPI.

$$\text{EVSI} = \text{EVofSI} = \text{EVDwithSI} - \text{EVDwithPriors} \quad (3.4)$$

But of course, typically not all the actual values for the required probabilities are given directly. There are then two interesting, and common, cases:

1. The probabilities of the possible outcomes given the possible reports are known, and the other probabilities are to be calculated. This is the outcome-given-report case and we consider it in the present section.
2. The probabilities of the possible reports given the possible outcomes are known, and the other probabilities are to be calculated. This is the report-given-outcome case and we consider it in the next section, §3.3.3.

Assume, for our present example, that we know the following probabilities:

1. $P(V) = 0.4$. We have this from the original problem statement.
2. $P(\bar{V}) = 1 - P(V) = 0.6$. Again, we have this as part of the basic problem.

3. $P(V|R(V)) = 0.7$. The probability of an actual visit by a parking authority person, given that the report from the street person says there will be such a visit, is 0.7. How do we know this? A student association at the local university has performed a careful scientific study on the past performance of the relevant predictions by street people and has arrived at this number.

Similarly, we have the following probabilities.

4. $P(\bar{V}|R(V)) = 1 - P(V|R(V)) = 0.3$. Read this as: The probability of not having a visit, given that the report says there will be a visit, is 0.3.
5. $P(V|R(\bar{V})) = 0.05$.
6. $P(\bar{V}|R(\bar{V})) = 1 - P(V|R(\bar{V})) = 0.95$.

With all this given, we still have two required probabilities undetermined: $P(R(V))$ and $P(R(\bar{V}))$. And using the fact that $P(R(\bar{V})) = 1 - P(R(V))$, we are left with one probability to compute. How do we do it?

It is useful here to invoke a very generally useful probabilistic identity. Let α and β be any two sets representing events (or sets of events). Recall that $\bar{\alpha}$ ("alpha bar" = α' or "alpha prime") is the set of events that obtains if any event outside α obtains (and similarly for other events, e.g., β and $\bar{\beta}$). Then, if α and β are disjoint, i.e., $\alpha \cap \beta = \emptyset$, then by the axioms of probability theory:

$$P(\alpha \cup \beta) = P(\alpha) + P(\beta) \quad (3.5)$$

(Note: This identity is intuitively correct and you should convince yourself it is in fact correct.) Next, we have a set-theoretic identity. For any sets α and β ,

$$\alpha = (\alpha \cap \beta) \cup (\alpha \cap \bar{\beta}) \quad (3.6)$$

(Note: This identity is also intuitively correct and you should convince yourself it is in fact correct.) Further, since $(\alpha \cap \beta) \cap (\alpha \cap \bar{\beta}) = \emptyset$,

$$P(\alpha) = P(\alpha \cap \beta) + P(\alpha \cap \bar{\beta}) \quad (3.7)$$

Recall the definition of conditional probability, Equation 3.1 above, repeated here as Equation 3.8.

$$P(\alpha|\beta) \stackrel{\text{def}}{=} \frac{P(\alpha \cap \beta)}{P(\beta)} \quad (3.8)$$

It follows that

$$P(\alpha|\beta) \cdot P(\beta) = P(\alpha \cap \beta) \quad (3.9)$$

Substituting into Equation 3.7 we get an important and general probabilistic identity:

$$P(\alpha) = P(\alpha|\beta) \cdot P(\beta) + P(\alpha|\bar{\beta}) \cdot P(\bar{\beta}) \quad (3.10)$$

Substituting our present values into Equation 3.10, we get

$$P(V) = P(V|R(V)) \cdot P(R(V)) + P(V|R(\bar{V})) \cdot P(R(\bar{V})) \quad (3.11)$$

Solving for $P(R(V))$ we get

$$P(R(V)) = \frac{(P(V) - P(V|R(\bar{V})))}{(P(V|R(V)) - P(V|R(\bar{V})))} \quad (3.12)$$

or

$$P(R(V)) = \frac{(0.4 - 0.05)}{(0.7 - 0.05)} = 0.54 \quad (3.13)$$

When we fold back the tree we find that if $R(V)$, i.e., if the report is that a visit is coming, then we plug the meter and the expected value of this fork is $0.54 \cdot -\$1.50 = -\0.81 . On the other hand, if the report is that a visit is not coming, then we do not plug the meter and the expected value of this fork is $0.46 \cdot (0.05 \cdot -\$15.32) = -\$0.35$. Taken together, the EVDwithSI is $-\$0.81 - \$0.35 = -\$1.16$. Since $\text{EVSI} = \text{EVDwithSI} - \text{EVDwithPriors}$, we have

$$\text{EVSI} = -\$1.16 - \$1.50 = \$0.34 \quad (3.14)$$

3.3.3 EVSI with Report-Given-Outcome Information

Suppose instead of the information we had in the previous case, §3.3.2, we have the following.

1. $P(V) = 0.4$. We have this from the original problem statement.
2. $P(\bar{V}) = 1 - P(V) = 0.6$. Again, we have this as part of the basic problem.

3. $P(R(V)|V) = 0.7$. The probability that the report from the street person says there will be visit by a parking authority person, given that such a visit actually occurs is 0.7. How do we know this? A student association at the local university has performed a careful scientific study on the past performance of the relevant predictions by street people and has arrived at this number.

Similarly, we have the following probabilities.

4. $P(R(\bar{V})|V) = 1 - P(R(V)|V) = 0.3$. Read this as: The probability that the report says there will not be a visit, given that there is a visit, is 0.3.
5. $P(R(V)|\bar{V}) = 0.05$.
6. $P(R(\bar{V})|\bar{V}) = 1 - P(R(V)|\bar{V}) = 0.95$.

Here, we need to calculate more quantities than in §3.3.2. In particular, we need to find

1. $P(R(V))$. (Note: $P(R(\bar{V})) = 1 - P(R(V))$.)
2. $P(V|R(V))$. (Note: $P(\bar{V}|R(V)) = 1 - P(V|R(V))$.)
3. $P(V|R(\bar{V}))$. (Note: $P(\bar{V}|R(\bar{V})) = 1 - P(V|R(\bar{V}))$.)

$P(R(V))$ is easy. Recall an identity from §3.3.2. Equation 3.10, reprinted below as Equation 3.15

$$P(\alpha) = P(\alpha|\beta) \cdot P(\beta) + P(\alpha|\bar{\beta}) \cdot P(\bar{\beta}) \quad (3.15)$$

Substituting in our known or assumed values, we have

$$P(R(V)) = P(R(V)|V) \cdot P(V) + P(R(V)|\bar{V}) \cdot P(\bar{V}) \quad (3.16)$$

or

$$P(R(V)) = 0.7 \cdot 0.4 + 0.05 \cdot 0.6 = 0.31 \quad (3.17)$$

Notice that

$$P(R(\bar{V})) = P(R(\bar{V})|V) \cdot P(V) + P(R(\bar{V})|\bar{V}) \cdot P(\bar{V}) \quad (3.18)$$

or

$$P(R(V)) = 0.3 \cdot 0.4 + 0.95 \cdot 0.6 = 0.69 \quad (3.19)$$

Now we consider how to calculate $P(V|R(V))$. To make this calculation it will be helpful, even essential, to have available to us another probabilistic identity, known as *Bayes's rule*. To derive a form of this rule, we begin with the definition of conditional probability, given above as Equation 3.1 and reproduced here as Equation 3.20

$$P(\alpha|\beta) \stackrel{\text{def}}{=} \frac{P(\alpha \cap \beta)}{P(\beta)} \quad (3.20)$$

Since $(\alpha \cap \beta) = (\beta \cap \alpha)$ we rearrange the right-hand side of Equation 3.20 to get

$$P(\alpha|\beta) = \frac{P(\beta \cap \alpha)}{P(\beta)} \quad (3.21)$$

But since, by the definition of conditional probability

$$P(\beta \cap \alpha) = P(\beta|\alpha) \cdot P(\alpha) \quad (3.22)$$

we substitute into Equation 3.21 and get:

$$P(\alpha|\beta) = \frac{P(\beta|\alpha) \cdot P(\alpha)}{P(\beta)} \quad (3.23)$$

Equation 3.23 is one version of Bayes's rule. And it is just what we need. Substituting our quantities (or their symbols) into Equation 3.23 we have

$$P(V|R(V)) = \frac{P(R(V)|V) \cdot P(V)}{P(R(V))} \quad (3.24)$$

or

$$P(V|R(V)) = \frac{0.7 \cdot 0.4}{0.31} = 0.90 \quad (3.25)$$

Notice as well that

$$P(\bar{V}|R(V)) = \frac{P(R(V)|\bar{V}) \cdot P(\bar{V})}{P(R(V))} \quad (3.26)$$

or

$$P(\bar{V}|R(V)) = \frac{0.05 \cdot 0.6}{0.31} = 0.10 \quad (3.27)$$

that

$$P(\bar{V}|R(\bar{V})) = \frac{P(R(\bar{V})|\bar{V}) \cdot P(\bar{V})}{P(R(\bar{V}))} \quad (3.28)$$

or

$$P(\bar{V}|R(\bar{V})) = \frac{0.96 \cdot 0.6}{0.69} = 0.83 \quad (3.29)$$

and that

$$P(V|R(\bar{V})) = \frac{P(R(\bar{V})|V) \cdot P(V)}{P(R(\bar{V}))} \quad (3.30)$$

or

$$P(V|R(\bar{V})) = \frac{0.3 \cdot 0.4}{0.69} = 0.17 \quad (3.31)$$

When we fold back the tree we find that if $R(V)$, i.e., if the report is that a visit is coming, then we plug the meter and the expected value of this fork is $0.31 \cdot -\$1.50 = -\0.47 . On the other hand, if the report is that a visit is not coming, then we again plug the meter and the expected value of this fork is $0.69 \cdot -\$1.50 = -\1.04 . Taken together, the EVDwithSI is $-\$0.47 - \1.04 , which, taking into account rounding errors, is $-\$1.50$. Since $\text{EVSI} = \text{EVDwithSI} - \text{EVDwithPriors}$, we have

$$\text{EVSI} = -\$1.50 - \$1.50 = \$0.00 \quad (3.32)$$

This information is not worth buying.

3.3.4 A Tabular Approach for Doing Certain Calculations²

The calculations we just illustrated using Bayes's rule are correct, but many find the algebra forbidding. So, here we suggest an alternative approach, or really representation since the underlying approach is the same, to problems calling for application of Bayes's rule.

Beginning abstractly, suppose we are given:

1. $P(\alpha|\beta)$
2. $P(\bar{\alpha}|\beta)$
3. $P(\alpha|\bar{\beta})$
4. $P(\bar{\alpha}|\bar{\beta})$
5. $P(\beta)$
6. $P(\bar{\beta})$

²Thanks to James D. Laing for suggesting the approach described in this section.

and we wish to find

1. $P(\alpha)$
2. $P(\bar{\alpha})$
3. $P(\beta|\alpha)$
4. $P(\bar{\beta}|\alpha)$
5. $P(\beta|\bar{\alpha})$
6. $P(\bar{\beta}|\bar{\alpha})$

We can express the information given in tabular form, as in table 3.1.

	β	$\bar{\beta}$
α	$P(\alpha \beta)$	$P(\alpha \bar{\beta})$
$\bar{\alpha}$	$P(\bar{\alpha} \beta)$	$P(\bar{\alpha} \bar{\beta})$
	$P(\beta)$	$P(\bar{\beta})$

Table 3.1: Tabular Approach to Bayes's Rule: Conditional Probabilities of Rows, Given Columns

Now, using just the information given in table 3.1, we can form a second table, table 3.2.

	β	$\bar{\beta}$	
α	$P(\alpha \beta) \cdot P(\beta)$	$P(\alpha \bar{\beta}) \cdot P(\bar{\beta})$	$P(\alpha \beta) \cdot P(\beta) + P(\alpha \bar{\beta}) \cdot P(\bar{\beta})$
$\bar{\alpha}$	$P(\bar{\alpha} \beta) \cdot P(\beta)$	$P(\bar{\alpha} \bar{\beta}) \cdot P(\bar{\beta})$	$P(\bar{\alpha} \beta) \cdot P(\beta) + P(\bar{\alpha} \bar{\beta}) \cdot P(\bar{\beta})$
	$P(\beta)$	$P(\bar{\beta})$	

Table 3.2: Tabular Approach to Bayes's Rule: The Given Information, Plus One Step

But if we just simplify the entries in table 3.2, using the definition of conditional probability, we get the joint probability distribution shown in table 3.3.

	β	$\bar{\beta}$	
α	$P(\alpha \cap \beta)$	$P(\alpha \cap \bar{\beta})$	$P(\alpha)$
$\bar{\alpha}$	$P(\bar{\alpha} \cap \beta)$	$P(\bar{\alpha} \cap \bar{\beta})$	$P(\bar{\alpha})$
	$P(\beta)$	$P(\bar{\beta})$	

Table 3.3: Table 3.2 Simplified: Joint Probabilities of Rows and Columns

Notice that in the right-hand column of table 3.3 we have two of the six items we are in search of, $P(\alpha)$ and $P(\bar{\alpha})$. In order to get the other four items we seek, we use the information in table 3.3 to get table 3.4.

	β	$\bar{\beta}$	
α	$P(\alpha \cap \beta)/P(\alpha)$	$P(\alpha \cap \bar{\beta})/P(\alpha)$	
$\bar{\alpha}$	$P(\bar{\alpha} \cap \beta)/P(\bar{\alpha})$	$P(\bar{\alpha} \cap \bar{\beta})/P(\bar{\alpha})$	

Table 3.4: (Almost) Final Table

But table 3.4 just simplifies to table 3.5, which contains the last four items we seek.

	β	$\bar{\beta}$
α	$P(\beta \alpha)$	$P(\bar{\beta} \alpha)$
$\bar{\alpha}$	$P(\beta \bar{\alpha})$	$P(\bar{\beta} \bar{\alpha})$

Table 3.5: Table 3.4 Simplified: Conditional Probabilities of Columns, Given Rows

3.3.5 A Note on Generalization

The concepts described in this section apply to more than the parking meter problem as presented. In fact, they apply to any decision tree, no matter how complex or whether or not the decision criterion is expected monetary value. Many of the formulas we used, however, might give you a different impression. Because we worked with only two branches from any node, all of the “general” formulas were expressed in terms of α , β , and $\bar{\beta}$. In the truly general case, these formulas may be expressed in terms of α , β_1, \dots, β_n . The key to generalization is this. The formulas so far rely upon the fact that $\beta \cap \bar{\beta} = \emptyset$ (β and $\bar{\beta}$ are mutually exclusive or disjoint) and $\beta \cup \bar{\beta} = \Omega$ (β and $\bar{\beta}$ are mutually exhaustive; together they cover all possibilities, represented by the universal set, Ω). So long as:

1. $\beta_i \cap \beta_j = \emptyset$ for all i, j where $i \neq j$
2. $\beta_1 \cup \beta_2 \cup \dots \cup \beta_n = \Omega$

the formulas may be generalized in a straightforward manner.

Moreover, the tabular calculation approach, described in §3.3.4, generalizes in the obvious ways to $\alpha_1, \dots, \alpha_m$ and β_1, \dots, β_n .

3.4 Utility Theory

Two questions:

1. Decision trees seem nice, but what if the outcomes aren't measured in dollars? What if outcomes are in lives saved, or hassle avoided or whatever?
2. Why should we decide based on expected value of dollars? Aren't we just taking expected values because we don't know what else to do with chance nodes?

These are good questions and it turns out that their answers are closely related.

Taking the second question first, it is indeed true that there is not any obvious real alternative to reducing chance nodes to their expected values. If we are going to reduce a chance node (or situation) to a single number, then expected value is the natural choice, if only for lack of an attractive alternative. Nor is there any attractive alternative to reducing chance nodes to single numbers, given that we want a numerical representation of the value of our best decision.

But if we are to take expected values, how should we measure the values of what it is we are taking the expected values of? (We're still on the second question, above.) Are dollars the right measure? Here's an example, due to Jacob Bernoulli and called Bernoulli's paradox, that shows that dollars are not always the right measure. Suppose you are offered the opportunity to participate in a game. The game works as follows. We begin with a fair coin (there are no tricks here and everything is as certifiably legit as you want). We flip the coin. If it comes up heads, you win \$2 and the game is over. If it comes up tails, you win nothing, but we continue to flip. If on the second toss the coin comes up heads, you win \$4 and the game is over, and if not we flip the coin a third time. All in all, we flip the coin until a head comes up and then we stop. You win $\$2^n$ where n is the number of flips it takes to get the first head. Nice game, but I'm not giving this away. You have to buy a ticket in order to play. Ask yourself what the most is that you would be willing to pay for such a ticket. Remember: you can have all the assurances you want that everything is done on the up and up.

Suppose (with no loss of generality) that you are willing to pay \$30 to buy a ticket for this game. Put another way, if offered a choice between getting \$30 or getting to play the game, you would be indifferent. If you had to pay \$31 to play the game, you would rather keep your money, but you would gladly pay only \$29 to play. So, you are indifferent between having \$30 and being able to play this game. Their values are about equal so far as you are concerned. Fine. Now, we know that you value the game at \$30, but what is the expected value of the game? It's infinite!

$$EV = \sum_{i=1}^{\infty} (1/2)^i \cdot \$2^i = \$1 + \$1 + \dots = \$\infty \quad (3.33)$$

So, your preferences are not consistent with expected monetary value. Moreover, since you cannot actually value any game infinitely, it is not possible to have any preferences at all that are consistent with expected monetary value, in this case. Hence, at least here, expected dollars cannot represent your preferences. Nor anyone else's really.

What are we to make of this? Bernoulli observed, as have others before and after him, that with increased wealth, added dollars (or whatever your favorite currency is) are individually valued less (by most people) than dollars gotten in relative poverty. In short, \$1,000,000 is worth a lot, but it is not worth as much as 1,000 · \$1,000. What, then, is it worth? Modern utility theory, the most directly relevant theory to the issues at hand, does not tell us. Rather, it tells us to expect that different people will legitimately differ on this question, and it tells us how to find out how someone values, say, \$1,000,000.

But this is, as it were, getting ahead of the game. Going back to our two questions, if we want to take expected values of gambles—chance nodes—for purposes of decision making and if dollars are not always the right way to measure outcomes, if we are to take expected values, then what do we do? We assume a sort of theoretical, abstract, certainly hypothetical, currency, called *utility*. Grounding all this are people's individual preferences. We should assume that references are more or less definite and concrete: people prefer this to that and are willing to act convincingly to demonstrate it. Utility is a numerical representation, or measure, of preference. The core idea of utility theory is that if a decision maker has preferences pertaining to some particular outcomes and those preferences meet certain specific conditions of rationality (more on this shortly), then there is a utility function on the

outcomes such that the decision maker prefers one lottery (gamble, chance node, etc.) over another if and only if the expected utility value of the one lottery is greater than that of the other. Put more simply, utility theory says that if you abide by the axioms (conditions of rationality) of utility theory, then there is always a way to measure the value of outcomes so that in the presence of uncertainty, taking the expected (utility) value of the decision trees is the right thing to do. If this is right, then indeed we have answered both of the questions that began this section.

For the interested, here are the four basic assumptions of utility theory. I present them following the excellent and accessible treatment by Kleindorfer, Kunreuther, and Schoemaker [9, Appendix A]. A notational convenience: suppose we have a lottery (gamble, chance node) in which outcome (or alternative) A occurs with probability p and outcome (or alternative) B occurs with probability $(1 - p)$. We shall represent this lottery as: $[p : A, B]$. Now to the four axioms of utility theory.

1. **Transitivity:** For any three outcomes, A, B, C , if a decision maker prefers A to B , and B to C , then the decision maker prefers A to C . Formally expressed: If $A \succeq B$ and $B \succeq C$, then $A \succeq C$, where $\phi \succeq \psi$ reads " ϕ is weakly preferred to ψ ."
2. **Continuity:** For any three outcomes, A, B, C , such that $A \succeq B \succeq C$, there is a probability, p , such that the decision maker is indifferent between B for sure and the lottery $[p : A, C]$. Formally, expressed: $B \sim [p : A, C]$, where " $\phi \sim \psi$ " reads " ϕ is judged to be equally as good as ψ ."
3. **Independence:** For any four outcomes, A, B, C , and D , suppose that $A \sim B$ and $C \sim D$, then for any p , $[p : A, C] \sim [p : B, D]$.
4. **Reduction:** For any alternatives, A and B , and for all probabilities, p, p_1 , and p_2 , $[p : [p_1 : A, B], [p_2 : A, B]] \sim [r : A, B]$, where $r = p \cdot p_1 + (1 - p) \cdot p_2$.

Nothing requires anyone to obey these axioms, but there is a certain attractiveness to them. They seem, at least to many people, to be acceptable normative principles of rationality. After all, can you really be rational if your preferences are not transitive?

There is very broad agreement that choice, if it is rational, must obey these principles *so long as the outcomes are not affected by another agent*. That is, when nature determines which outcomes occur, then expected utility is, it is generally agreed, the right set of principles for rational choice. If, however, other agents may have a hand in determining which outcomes occur, then we have moved into the realm of game theory and the broad consensus on expected utility no longer obtains. In fact, the problem of characterizing rationality in game-theoretic contexts remains at the frontier of contemporary research.

But these are deeper issues and we must reluctantly move on.

3.4.1 Eliciting a Utility Function

If we can measure outcomes in utilities, instead of say dollars, then we can be confident that using decision trees and the expected value criterion has a solid basis. The analysis of the decision trees proceeds just as explained above, except that values—of outcomes, of decision nodes, and of chance nodes—are denominated in utilities. So if we can measure outcomes in terms of utilities we are in a happy circumstance. How, then, *do* we actually go about making those measurements?

What we need is to transform the values of outcomes measured in dollars, oranges, square feet of living space, and so on, into the coin of utility. We do this by finding a utility function that takes a quantity (of dollars, oranges, square feet of living space, or whatever) as input and returns a number representing the utility of that quantity. For reasons that are beyond the scope of our present purposes, we can set the utilities of any two outcomes as we please.³ For convenience, we assume throughout that the decision maker's preferences are monotone functions of the underlying argument and we set the utility of the worst outcome considered to be 0 and the utility of the best outcome considered to be 100.

³Briefly, each person's utility function (in a given context) is unique up to a positive linear transformation. That is, if $U(x)$ is a utility function, then for any other utility function, $U'(x)$, $U'(x)$ is a positive linear transformation of $U(x)$ if and only if $U'(x) = \alpha + \beta U(x)$, for some numbers α and β , where $\beta > 0$. Temperature scales are positive linear transformations of each other. For example, $C = (5/9) \cdot (F - 32)$. With utilities, as with temperature scales, we can establish arbitrarily an origin and a unit of measurement. Thus, with utility we can use $[0, 1]$ or $[0, 100]$ or $[-3.2, \pi]$ or whatever is convenient.

The utilities of all other outcomes are assessed subjectively; we ask the decision maker several questions about his or her preferences and we estimate a utility function from the answers given. Typically, we assume a function of a particular form, we ask several questions, and we find and use the best fitting function of the assumed form. We can, however, elicit utilities for particular outcomes without making any assumption about the form of the underlying utility function. There are many ways of doing this. Let us look at one.

For the sake of concreteness, we will advert to the parking meter problem. Suppose that instead of working with expected dollars, we decide to perform the analysis in terms of expected utilities. Our best outcome is \$0.00, not plugging the meter and not getting a ticket. Set the utility of this outcome to 100: $u(\$0.00) = 100$. Our worst outcome is $-\$15.32$, getting a ticket after not plugging the meter. Because \$15.32 is such an awkward number and because we would like to build in some flexibility (What if the postal rates go up?), we will set the utility of $-\$18.00$ to 0: $u(-\$18.00) = 0$. Given this, we can construct a lottery for which we can calculate its expected utility: $[p : u(\$0.00), u(-\$18.00)] = [p : 100, 0]$. The expected utility of this lottery is: $p \cdot 100 + (1 - p) \cdot 0 = p \cdot 100$. So, if $p = 0.5$, the expected utility is 50.

We are now ready to ask our decision maker a question: "Dear Decision Maker, Suppose that you are presented with the lottery,

$$[0.5 : \$0.00, -\$18.00] \quad (3.34)$$

This is not exactly a nice situation, since you stand to gain nothing and could lose \$18.00. The chance is even either way. Suppose, further, that you are stuck with this lottery, unless you can get someone to take it away from you. What is the most you would pay someone to take this lottery off your hands?" Utility theory says nothing about how our decision maker should answer. What is right is a matter of personal preference. What utility theory offers is this: if the decision maker is willing to answer a short series of such questions, then a reliable model of the decision maker's preferences can be built and used in a large number of independent cases. Since these cases may be quite complex, it is reasonable to hope that the model, built from the decision maker's judgments in fairly simple cases, will be more accurate on complex cases than the decision maker's own direct judgments. Such are the hopes for this sort of thing.

The decision maker—who prefers more money to less—has a rational right to give us any answer between \$0.00 and -\$18.00. Suppose the decision maker reasons as follows: "I'd like to pay as little as possible and in a market situation I would negotiate and shop vigorously, but I only have \$20.00 with me today and if I pay more than \$12.00, I won't be able to have lunch and ride the bus home. So, that's the most I would pay. Anything more and I'll take my chances." What the decision maker is telling us is that he or she is indifferent between losing \$12.00 for sure and facing the lottery, which has a utility of 50. So, $u(-\$12.00) = 50$. Note that the expected dollar value of our lottery is -\$9.00, so our decision maker is willing to pay more than the expected dollar loss in order to get rid of this gamble. In such conditions, we say that the decision maker is *risk averse*. If the decision maker were only willing to pay \$7.00 to be rid of the lottery, he or she would be *risk seeking*. Finally, if the decision maker's expected utility is identical with the expected monetary value of the lottery, i.e., if the most the decision maker would pay to be rid of the lottery is \$9.00, then we say the decision maker is *risk neutral*. If the decision maker is risk neutral, then using the utility function instead of, here, the dollar amounts in the decision tree will not affect the recommended decision.

We are now in position to fit a functional form to our elicited point. If the decision maker is risk neutral, then we would have a straight-line utility function, as follows

$$u(x_i) = 100 \cdot \left(\frac{(x_i - w)}{(b - w)} \right) \quad (3.35)$$

where x_i is the value (here, dollar value) of the outcome in question, b is the value of the best outcome under consideration (here, $b = \$0.00$) and w is the value of the worst outcome under consideration (here, $w = -\$18.00$). Thus, in the risk neutral situation,

$$u(-\$9.00) = 50 = 100 \cdot \left(\frac{(-\$9.00 - (-\$18.00))}{(\$0.00 - (-\$18.00))} \right) \quad (3.36)$$

We can generalize Equation 3.35 in a simple manner, and accommodate risk aversion and risk seeking. Let

$$u(x_i) = 100 \cdot \left(\frac{(x_i - w)}{(b - w)} \right)^a \quad (3.37)$$

where α is a risk aversion factor. When $\alpha = 1$, the resulting utility function is risk neutral. When $0 < \alpha < 1$ the utility function is risk averse, and when $\alpha > 1$ the function is risk seeking.

Suppose we choose to model with Equation 3.37, what has our decision maker told us about the value of α in this problem? We have

$$50 = 100 \cdot \left(\frac{(-\$12.00 - (-\$18.00))}{(\$0.00 - (-\$18.00))} \right)^\alpha \quad (3.38)$$

which reduces to

$$\alpha = \frac{\ln(1/2)}{\ln(1/3)} \approx 0.63 \quad (3.39)$$

Thus, our utility function for this (hypothetical) decision maker, for dollars over the range \$0.00 to -\$18.00 is

$$u(x_i) = 100 \cdot \left(\frac{(x_i - (-\$18.00))}{(\$0.00 - (-\$18.00))} \right)^{0.63} \quad (3.40)$$

Applying this to the parking meter problem we find that the expected utility of not plugging the meter is

$$0.4 \cdot u(-\$15.32) + 0.6 \cdot u(\$0.00) = 0.4 \cdot 30.12 + 0.6 \cdot 100 \approx 72$$

On the other hand the utility of plugging the meter is

$$u(-\$1.50) \approx 95$$

So, again it is best to feed that meter.

Finally, note that if the decision maker is sufficiently risk seeking, the expected utility of not plugging the meter will be higher than the expected utility of plugging the meter. We leave it as an exercise to the reader to determine just how risk seeking our decision maker would have to be to prefer leaving the meter unplugged.

3.4.2 Comments and Warnings

A short list of points will serve our purposes:

1. The elicited utility function is local to the model. If a broader range of dollars is at stake, the function must be reassessed. That is one reason it is usually a good idea to assume, as we did, a slightly wider range of outcome values than is necessary for immediate purposes.
2. The functional form we used is but one of many. The significant differences are typically minor, for practical purposes, among the functional forms that are—as ours is—constantly increasing (or decreasing).
3. Interpersonal comparisons of utility, e.g., Susan's utilities compared to Maggie's utilities, are theoretically nonsense, however much we would like it to be otherwise. (Although, e.g., Susan's utility function is a positive linear transformation of Maggie's, the parameters in the transformation (α and β) are arbitrary and unknowable.)
4. Utility theory is very widely accepted as normatively sound. Still, it has its critics, as well as its paradoxes and anomalies. It should be used as any other tool, with caution, common sense, and lots of post-evaluation (especially sensitivity) analysis.
5. Elicitation of utility functions is something of an art. If you find that the functions elicited are multimodal or in other ways odd-looking, get some expert advice before relying heavily on the associated model.
6. It is easy, and quite common, during the elicitation process for decision makers to give answers that do not fit the chosen functional form exactly. When this happens, you will need to fit the functional form approximately, with some error. Check to see that the errors are fairly small and not all on one side.

3.5 Multiattribute Utility Theory (MAUT) Models

Another question:

- Expected utility theory seems useful in cases in which uncertainty is present and the outcomes can be measured in terms of dollars or oranges or square feet of living space or other simple denominations. But most

important decision problems involve outcomes that are not so simple. Outcomes may have many aspects to them. For example, in choosing a supplier, one supplier may be better on cost and worse on quality and delivery time. How are we to model these sorts of tradeoffs with expected utility theory?

The question is asking about what, in the jargon of the utility theory literature, are called *multiattribute decision problems*, in which the outcomes have associated with them several attributes, or aspects or dimensions, of value. Indeed, such problems are the norm. Just about anything one buys has multiple dimensions of value. In a car, the attributes include reliability, mileage, safety, and resale value. In an apartment, the attributes include living space, quality of the neighbors, commitment of the landlord, amenities in the neighborhood, location, and much else.

As usual, we need to reduce things to a single number and proceed as before. So, we will represent the values of multiattribute outcomes to single (utility) values, and then proceed as before. There are basically two strategies. First, we might translate measures on each attribute to a common scale, such as dollars, and then translate that scale to utility. When this is natural, it is a perfectly sensible thing to do, but often it is not, and one should use the second strategy: create a multiattribute utility function.

In the second strategy, we develop distinct utility functions for each attribute of interest. Such utility functions are called *unidimensional utility functions*, in order to distinguish them from multiattribute functions. Earlier, in the parking meter example, we saw how a single unidimensional function could be developed. In the multiattribute case, we repeat this process for each attribute. Once the individual (unidimensional) utility functions have been obtained, we then combine them mathematically to create a multiattribute utility function.

The simplest, and most commonly used, form of combination is the *weighted average*, or *linear or additive*, model:

$$U(X_i) = k_1 \cdot u_1(x_{i,1}) + k_2 \cdot u_2(x_{i,2}) + \dots + k_n \cdot u_n(x_{i,n}) = \sum_{j=1}^n k_j \cdot u_j(x_{i,j}) \quad (3.41)$$

A word on the notation. We use lower-case *u*s, possibly subscripted, to represent unidimensional utility functions, and upper-case *U*s, possibly subscripted, to represent multiattribute utility functions. Thus, the *u_j*s are uni-

dimensional utility functions and there is one for each of the n dimensions or attributes at hand. The k_j s are relative importance weights. They range from 0 to 1 and must sum to 1. A multiattribute outcome is represented by a vector, $X_i = \langle x_{i,1}, x_{i,2}, \dots, x_{i,n} \rangle$, in which an individual entry, $x_{i,j}$, is the score, or measure, or description on dimension j for outcome i .

The weighted average model, Equation 3.41, has much to recommend it. It is simple, it is intuitive, and it is robust. But it does make certain assumptions that may be incorrect and could invalidate the model. For present purposes, we will say little about this issue except to note that the model requires a kind of independence that is easily checked (more on this shortly).

In order to build a specific weighted average model, Equation 3.41, there are several things we need to know. Here is a systematic list, with discussion. The list is an augmented version of the SMART (simple multiattribute rating technique) procedure from Ward Edwards [6, 12].

1. Identify whose values are to be modeled.

Whose decision is it? In whose name is the decision being made? This is an obvious and elementary first step, but it is one often neglected.

2. Discover the purpose of the modeling exercise.

What goals are relevant? Why are we doing this? Answers to these questions will serve to clarify and focus the exercise on the issues really at hand.

3. Determine the alternatives to be evaluated.

A reasonably precise determination of the relevant options, or entities, to be evaluated is important for many reasons, especially for determining what the relevant dimensions of value are.

4. Identify the relevant dimensions, or attributes, for evaluating the alternatives.

This is a crucial and sensitive step. Avoid having too many attributes. A model with more than a dozen attributes is not likely to be successful or well designed. Choose attributes that can be measured in a meaningful and practicable way. Determine the values over which the individual attributes will range. For example, if the attribute can be

measured in dollars, choose a range of dollars that includes the values likely to be encountered in the options to be evaluated

5. Rank the attributes in order of relative importance.

Ask the following question:

Suppose we have n attributes. Consider n distinct options or alternatives, X_1, X_2, \dots, X_n . Now imagine that we create the following hypothetical options for the purpose of the analysis. Option X_1 has the top possible scores on each of the n attributes, except attribute 1, on which X_1 has the lowest possible score. Option X_2 has the top possible scores on each of the n attributes, except attribute 2, on which X_2 has the lowest possible score; and similarly for the other alternatives. How do you rank these n alternatives in order of preference?

From the resulting ranking we can read off the relative importances of the attributes. If X_i has the highest (second highest, third highest, ...) rank, then attribute i has the lowest (second lowest, third lowest, ...) relative weight.

Comment: The questioning can proceed in other ways, and it is advisable to try them as well. For example:

Suppose we have n attributes. Consider n distinct options or alternatives, X_1, X_2, \dots, X_n . Option X_1 has the **lowest** possible scores on each of the n attributes, except attribute 1, on which X_1 has the **highest** possible score. Option X_2 has the **lowest** possible scores on each of the n attributes, except attribute 2, on which X_2 has the **highest** possible score; and similarly for the other alternatives. How do you rank these n alternatives in order of preference?

From the resulting ranking we can read off the relative importances of the attributes. If X_i has the highest (second highest, third highest, ...) rank, then attribute i has the highest (second highest, third highest, ...) relative weight.

The final rankings from these two question forms should be the same. If they are not, this probably indicates that the linear model is inappropriate.

Note well: The question is (properly) asked with direct reference to the highest and lowest possible scores on each dimension. Setting these scores is done in step 3, above. We would expect that different ranges on an attribute will result in a different relative weight for that attribute. If, for example, the range of possible scores on an attribute is small, then it is likely that the relative weight on that dimension should also be small.

6. Obtain ratio estimates of the relative weights, relative to the least important dimension.

Give the lowest-ranked (least important) attribute a 10. Now ask: For the second-lowest-ranked attribute, what should its score be? Continue in this fashion for all the remaining attributes.

7. Normalize the ratio estimates of the relative weights.

We obtain the individual k_i values (which, recall, range from 0 to 1 and sum to 1) by dividing each estimated score (from step 6) by the sum of all the score estimates.

Note: This is also a good technique to use in spreadsheet implementations, since it allows us to do sensitivity analysis on any k_i or combination of k_i s.

8. Obtain a unidimensional utility function for each attribute (for the scale and range determined previously).

We have already seen how to do this in our discussion of the parking meter problem.

Note: We assume that each utility function ranges from 0 to 100, but all that is really necessary is that all the utility functions have the same range.

9. Score each attribute of each alternative.

Recall that each alternative may be thought of as a vector of descriptions on separate dimensions

$$X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,n}\}$$

Obtain those $x_{i,j}$ descriptions. (Note: the $x_{i,j}$ values obtained in this step need to be numbers, e.g., numbers of dollars. This does not preclude assessing the values of attributes that are not naturally measured numerically. Far from it. One merely needs to develop, say, a phrase-anchored scale that maps verbal descriptions to numbers. The details are beyond the scope of our present goals.

10. Calculate the utilities of the various options using a weighted average model.

At this point, we have all the information needed to calculate the utilities of the alternatives using the linear additive model, Equation 3.41. So, we make the calculations.

11. Perform post-evaluation analysis and decide.

Prima facie, the best alternative is the alternative with the highest calculated expected utility. Before making a final choice, however, you should examine carefully what the model is saying. How does the ranking of alternatives change with slight changes in the relative weights (k_i s), scores on the attributes, and unidimensional utility functions? If the changes make sense, then that is evidence the model is valid. If the model also exhibits a fair degree of robustness, then you may be able to decide with confidence.

3.6 Comments on the Use of Decision Analysis

There is very much more to this topic, but we have here a serviceable introduction. The next step is to build and implement some models, and we take that up in the sequel.

Some general comments on decision analysis.

1. MAUT models are often, even usually, used without decision trees. When they are used with decision trees, it is appropriate to model multiattribute outcomes, get their utilities from the MAUT model, and proceed in the usual manner. Of course, this adds to the complexity of the overall decision analysis and mandates careful post-evaluation analysis.
2. These (decision trees and MAUT models) are good methods and should be used more. In fact they are not used as nearly as much as they should. Why?
 - (a) People tend to think that they are experts in making decisions—but they are not!
 - (b) Computertization is needed for all but the smallest models. Good packages are not readily available and well understood. In the sequel, we shall see how to build decision tree and MAUT models using spreadsheets.
 - (c) It takes time and effort to build decision analytic models, and the basic techniques are unfamiliar to many people. Often, it is best to hire special consultants to help structure the decision and elicit information.
 - (d) Extensive use of *subjective data*, data elicited from individuals. This bothers some people (as it should). Defense: this is the best we can do; and we then do sensitivity analysis. At least the subjective judgments are up front and open.
3. A very important value: structuring the decision. Decision analytic models are helpful, if only for encouraging the deliberate, reflective structuring of the problem. Because the resulting models are public, they can be inspected, queried, and generally poked by all relevant stakeholders.

3.7 Bibliographic Notes

Recommended reading: from Bazerman [1], chapter 2, “Biases,” of *Judgment in Managerial Decision Making*.

There are many excellent books and articles on decision analysis. Among them are: [2, 7, 11, 12] For information on, and a source of comfort for, the linear model, see [5].

Chapter 4

Notes on: Decision Trees with Spreadsheets¹

4.1 Introduction

Our topic now is implementing decision analysis with spreadsheets. What do we want from an implementation?

1. Calculate the expected value of the tree
2. Indicate the optimal decision path
3. Facilitate sensitivity analysis
4. Indicate invalidities (at least certain kinds of them)
5. Be easy to understand and work with (highly documented!)
6. Be easy to modify and maintain

How are we going to do it? First, here are the essential spreadsheet skills with which you need to be familiar to perform the exercise of building a decision tree in this way.

1. Layout and design of Excel (e.g., worksheets and workbooks)

¹File: dt-dtree-with-as. Revised: 951222, 951129, 951022. From: (MISNotes-dtreewith-as. Revised: September 20, 1995.)

2. Presentation and formatting of Excel objects and entities (e.g., naming worksheets, using attractive formatting, coloring things, setting protection)
3. Absolute and relative addressing
4. Formulas (e.g., IF, MAX)
5. Drawing and graphics
6. Charts
7. Goal seeking, data tables

Now to building a decision tree DSS in Excel, version 5.0 or later. (We focus on a particular spreadsheet product for concreteness, but the general lessons and principles apply to essentially all spreadsheet products.)

The usual strategy—with any spreadsheet product—is to focus on making the display look like a decision tree. Problems:

1. This is really forcing things in a spreadsheet. Lots of work.
2. Hard to maintain.
3. Hard to manipulate

Of course, when presenting results of an analysis it will almost always be useful to present the relevant decision trees to your audience. But, the decision tree display should be driven by a primary representation outside the tree.

Another way: reduce to tabular form and work that way. Think of elements of the problem and organize this way, with elements of a common type on a single worksheet (again: assume we are using Excel 5).

1. Rename "Sheet1" as "Introduction." Put comments, a table of contents, and other overview text and information here.
2. Rename "Sheet2" as "Decision Tree." Later, draw the decision tree on this sheet and fill in the values with names referring to other parts of the workbook.

3. Rename "Sheet3" as "Input Parameters." Put all input parameters on this sheet, named and laid out as described below, §4.3.
4. Rename "Sheet4" as "Interior Results." Put all formulas for intermediate results on this sheet, named and laid out as described below, §4.4.
5. Rename "Sheet5" as "Chance Nodes." Put all chance nodes, represented as tables, on this sheet, named and laid out as described below, §4.5.
6. Rename "Sheet6" as "Decision Nodes." Put all decision nodes, represented as tables, on this sheet, named and laid out as described below, §4.6.

This will serve as a basic template for us, to be revised and expanded later. We will begin by working on the parking meter problem using the decision rule of expected dollars. Utilities come later.

4.2 Introduction to Decision Tree Implementation

Below, in §§4.3–4.6, we will give some essential information on how to implement a decision tree program in a spreadsheet for the parking meter problem. What follows should be seen as a guideline. It is meant to be helpful, not to be complete. In an actual implementation we would need to add more, for example, alerters to indicate the optimal decisions and to indicate invalid data (parameter) values.

Think, for starters, of your program as being divided up into a series of tables, as follows. You should be able to figure out appropriate locations in the worksheets for the various names and labels.

And, recall the basic tree for our parking meter problem, Figure 4.1.

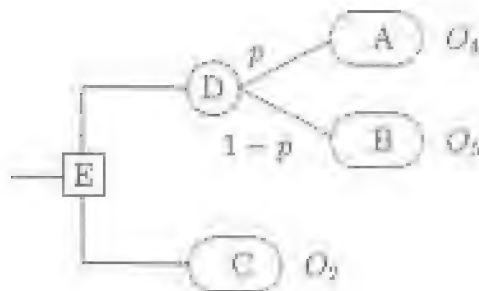


Figure 4.1: Decision Tree for the Parking Meter Problem

4.3 “Parameters” Sheet

Table 4.1, lays out the input values for the model’s parameters.

Value	Name	Meaning
-15.32	AA	Get a ticket
0.00	BB	Don’t plug meter and don’t get a ticket
-1.50	CC	Plug meter
0.4	PP	Relative probability of getting a ticket
0.6	QQ	Relative probability of not getting a ticket

Table 4.1: Parameters for the Parking Meter Problem

Note: the symbols should all be defined as names for the values to their left. Also, AA, BB, and CC are outcome values. It is a good idea to label them as such in some way. Finally, PP and QQ are relative probabilities only. As parameters, the values of either or both may be changed in this table, e.g., during sensitivity analysis.

Admittedly, the AA, BB, etc. notation and naming scheme here could be improved. You should think of mnemonic names, e.g., `getticket` instead of AA, and `relprobticket` instead of QQ.

4.4 “Formulas” Sheet

Table 4.2 which presents the formulas used in this model (there happens to be only two formulas used). These formulas, named “probticket,” and “probnoticket,” calculate the absolute probabilities of getting (resp., not getting) a ticket, if the meter is not plugged.

Value	Name	Meaning
$=PP/(PP+QQ)$	probticket	The probability of getting a ticket
$=QQ/(PP+QQ)$	probnoticket	The probability of not getting a ticket

Table 4.2: Formulas for the Parking Meter Problem

4.5 “Chance Nodes” Sheet

In the parking meter problem we have only one chance node, D, which will be named DD for the sake of Excel. See Table 4.3.

Daughter Node	Probability	Value
AA	=probticket	=AA
BB	=probnoticket	=BB

Table 4.3: Chance Nodes: Layout and Key Formulas

With such a table, pick a convenient cell and put into it a formula that calculates the expected value. Hint: use the sumproduct function. Name this cell DD.

4.6 “Decision Nodes” Sheet

In the parking meter problem we have only one decision node, E, which will be named EE for the sake of Excel. See table 4.4.

Daughter Node	Value
CC	=CC
DD	=DD

Table 4.4: Decision Node EE: Layout

Note that the formula, =CC, simply refers to the value of the parameter, CC, by the magic of naming. Further, we will name the expected value of the DD node as DD; for then the formula, =DD, will refer to the expected value of the CC node. (Again. This naming convention could be improved and you may want, e.g., to begin the name of each decision node with a “D” and each chance node with a “C” followed by some indication of what the daughter nodes are.)

But we need two more things from the EE node implementation:

1. The expected value of the node. This is just the maximum (use the =max function) of the values of the daughter nodes. You should create a formula for this in a cell and name the cell EE.
2. The name of the daughter node corresponding to the optimal decision. There are several ways to handle this. Here’s one. Add a new column to the decision table, as follows. Here, we are assuming that the

Daughter Node	Value	Chosen?
CC	=CC	=IF(MAX(\$E\$5:\$E\$6)=E5,1,0)
DD	=DD	=IF(MAX(\$E\$5:\$E\$6)=E6,1,0)

Table 4.5: Decision Node EE: Expanded Layout to Include the Optimal Choice

“Value” column is in column E of the worksheet, that the =CC formula is in cell E5, and that the =DD formula is in cell E6. Now, in some convenient cell, say cell G4, put in the formula =D5, and name this

cell, say, eebeetdaughter. Then, write a macro that sorts the table's range, D5:F6, (or better, uses a name given to the table) in descending order. After you run the macro, cell G4 will contain the name of the best daughter cell from decision node EE. Later, you can assemble all such macros and attached them to a button.

4.7 Discussion

Much more can be done. Here are some suggestions:

1. On the "Decision Tree" worksheet you can draw the decision tree and label its elements using the existing names on the other worksheets. That way, when you alter the assumptions, e.g., the value of BB, the drawn decision tree can be more or less automatically updated.
2. On the "Introduction" worksheet, you can describe the DSS that follows and how to work with it.
3. This is a small DSS. What could you do by way of formatting and presentation to make it friendlier, easier to use, and easier to maintain? Example: Protect the worksheets from changes by users, except for the cells holding parameter input values, and color these green.
4. You can do sensitivity analysis by asking what-if questions. Go to the "Parameters" worksheet and change values. See how the decision changes, if at all. You can also do more sophisticated analyses, using data tables, graphics, and goal seeking commands.
5. You can alter what has been done so far to create a more elaborate tree.
6. You can alter what has been done so far, and measure the outcomes in utilities, rather than dollars.

And there is much more. Be creative!

Appendix A

Visual Basic for Applications: A Brief Tutorial¹

A.1 First Steps

Visual Basic for Applications (VBA) is the macro language for Excel. It closely resembles Visual Basic, an independent language from Microsoft, and is used as the macro language for Microsoft Access and Word. In what follows, we will be talking for the most part about Visual Basic for Applications as it applies to Excel. We will feel free to call it VBA, EVB, Visual Basic, VB, etc., so long as the context makes confusion unnecessary.

Macros consist of one or more VBA code chunks. These code chunks—*procedures*—are either *functions* or *subroutines*. Here are some simple examples.

```
' Here is a simple function. Use as any other Excel function.
Function bob(x)
    bob = x ^ 2 + 3.34
End Function
' Here is a simple VBA sub.
Sub ted()
    MsgBox "Hello, world!"
End Sub
```

¹File: dt-vbatutor. Created: 951128, from VBTUTORF.DOC. Revised: 951222.



Figure A.1: Help Menu for Excel 5.0

Note: comments begin with a single quote:

- ' Everything afterwards in the line is ignored.

In Excel, VBA macros reside on special workbook sheets, called *modules*. To make a macro, one may simply create a new macro module and type in the functions and procedures. More on this shortly.

Information about VBA is published in many readily-available sources. Both Microsoft and third-parties publish extensive reference manuals and how-to books for VBA. In addition, VBA closely resembles Visual Basic and there is a large literature on that. For good online help on VBA in Excel, explore "Programming with Visual Basic" in the "Contents" window of the MS Excel help facility (see Figure A.1). We are assuming in these notes that the reader will do this.

A.2 Second Steps

A.2.1 Recording Macros

Macros (VBA procedures) can be recorded. Use Record Macro under the Tools menu. After selecting Record New Macro, you will be prompted for the name of this new macro. Either give it a new name, or accept the default. A small window will then appear with a stop button in it. You click the stop button when you are done recording your macro. First, however, perform as usual some action in the workbook, e.g., copy one range of cells to another place. When you are done, stop the macro recorder by clicking the stop button. In sum, there is a four-step process to record a macro:

1. Start the macro recorder.

Do this by selecting the menu: Tools / Record Macro / Record New Macro.

2. Name the macro.

You will be prompted for a name and may accept the default presented by Excel, e.g., Macro1. Once you have done this, a window appears with a button for stopping the recording of the macro.

3. Record the macro by performing normal activities in the workbook.

It is wise to plan these out before starting to record.

4. Stop recording the macro.

Do this by clicking the stop macro button.

This creates VBA code in a (usually new) module sheet, which Excel will call Module1 or some such thing. Module sheets reside with the other sheets of the workbook. As with the other sheets, you click on the tab to view the module sheet. When it appears, you will see VBA code against a blank background. While worksheets present spreadsheets (arrays of cells), macro sheets present text editors. Thus, you can examine and edit the VBA code.

Notice, in particular, a couple of things with regard to your new macro module sheet. First, macro sheets come with a *context-sensitive* text editor. For example, comments (lines beginning with an apostrophe) come out green

(by default) and reserved words come out blue and get capitalized automatically. Second, the new macro that you just recorded is a Sub, rather than a Function.

Recording macros and examining the results is a good way of learning about VB, but it takes you only so far. We need to go further.

A.2.2 Assigning a Macro to a Button or Graphic Object

In order to run (or execute) a Sub macro, including macros created with the Record New Macro facility in Excel, one can choose to assign the macro to a graphic object that can call the macro. Assigning a macro to a button or graphic object is easy. For a previously-existing object, select it (e.g., hold down the Ctrl key and click on the button or graphic object), then choose Assign Macro... from the Tools menu. You will be prompted with a list of existing Subs and you make your choice from the list. That done, you may now simply click on the graphic object and Excel will call the macro and cause it to be executed.

Note: Typically, you will want to create a new button and assign the macro to it. Use Create Button from the Drawing icon and draw a new button. Excel will automatically prompt you to assign a macro.

A.2.3 Functions versus Subs

VBA functions return values (one value each), but cannot take actions otherwise. VBA subs (subroutines) do not return values, but can take actions. (However, VBA subs can set the values of variables and these variables may be accessed by other procedures.) VBA functions, once defined in a macro sheet, may be used in worksheet cells just as any of the functions Excel has built into it.

Functions and subs may call one another, thus you may create very complex programs in VBA. We will discuss that later. First things first. Now, let's look at variables in VBA.

A.3 Variables

A.3.1 The Very Basics of Variables

Here is a simple example involving VBA variables:

```
' Here's an example of two variables in use, along with
' a For...Next... control loop
```

```
Sub variableExample1()
    ' Assign the number 3 to the variable, MyVariable
    ' Note: You make up your own, mnemonic,
    ' names for your variables

    MyVariable = 3
    For i = 1 To MyVariable
        MsgBox "Showing and counting: " & i
    Next i
End Sub
```

The two variables are: `MyVariable` and `i`.

Such program variables are used extensively in this sort of programming. Variables hold values and their values may change during program execution. Basically, you make computations and assign the results to variables. Then you make new computations, based on the assigned values of these variables, and you assign the results to other variables. And on and on.

A.3.2 Variables Have Data Types

Some variables are for holding numbers, some for text, some for dates, and so on. VBA has a special type of variable, called the *variant* type. It can hold about anything, but in general you should avoid being so loose.

The main data types in VB are

1. Boolean. Values: True or False
2. Integer. Values: -32,768 to 32,767
3. Long (integer). Values: -2,147,483,648 to 2,147,483,647

4. Single (single precision floating point). Values: [lots]
5. Double. Values: [lots more than singles]
6. Currency. Values: [lots]
7. Date. Values: January 1, 0100 through December 31, 9999
8. String. Values: 0 through 65,535 characters
9. Variant. Values: Any numeric value thru Double or any character text

You set the data type of a VB variable by declaring it. But, if you don't declare the data type for a variable (as in the `variableexample1` procedure, above), then the default is that the variable is of type variant.

Within a procedure, you may declare variables with the `Dim` (dimension) statement.

```
' Now here's variableExample1 again, but
' with the variables properly declared
Sub variableExample2()
    ' Assign the number 3 to the variable, MyVariable
    ' Note: You make up your own, mnemonic,
    ' names for your variables
    Dim MyVariable As Integer
    Dim I As Integer
    MyVariable = 3
    For I = 1 To MyVariable
        MsgBox "Showing and counting: " & I
    Next I
End Sub
```

A.3.3 Local and Global Variables

Variables declared this way (explicitly in a procedure with `Dim` or as variant by default) are *local* to the procedure. That is, you can't refer to them—use their names and get their values—in other procedures. In fact, as illustrated in `variableexample1` and `variableexample2`, above, you can actually reuse the same variable names in different procedures. When you do this, you are

really working with different variables, which happen to have the same names. (Advice: except for counters, like `I`, and explicitly temporary variables, e.g., `mytemp`, don't do this.)

Point of style: It is normally considered good programming practice to declare all your variables explicitly. Why? In Visual Basic, you can enforce this by declaring

`Option Explicit`

in the declarations section of each code module. (The declarations section of a module is the space before the first procedure—i.e., at the top.) You should do this. Then, when VB encounters a variable that hasn't been declared, VB generates an error message. This may initially be irritating, but it's a very good idea in the long run, since it prevents otherwise undetected errors.

The scope of a variable need not be limited to being local, however. In VBA in Excel, the scope of a variable may be the procedure in which it is declared (in which case we say it is local), the module in which it is declared, or the entire workbook.

When the scope is to be local (within procedure only), declare variables at the beginning of the procedure with the `Dim` statement. (See also in the Help facility: the `Static` statement.) See examples above, procedures `variableExample1` and `variableExample2`.

When the scope of a variable is to be the module in which it is declared, declare the variable at the top of the module (in the declarations section), using `Dim`. (See also in the Help facility: the `Static` statement.)

When the scope of a variable is to be the entire workbook, pick a module, and declare the variable in the declarations section using `Public` (cf., `Global`).

Here's an example:

```
' Each module begins with a declarations section, the
' portion at the top, before the procedure declarations
' begin.

' Declare explicit data type checking
Option Explicit
Public MyVar As Integer
```

```

Sub publicExample1()
    MyVar = 17
    MsgBox "We're in publicExample1 and MyVar = " & MyVar
    'publicExample2
End Sub

```

```

Sub publicExample2()
    MsgBox "We're in publicExample2 and MyVar = " & MyVar
End Sub

```

Note: "Module-level variables remain in existence while Visual Basic is running until the module in which they are defined is edited" (Visual Basic User's Guide, Microsoft Excel 5.0, p. 121). So play around with this example and see how this stuff works.

A.3.4 Reading from an Excel worksheet into an Excel Visual Basic Variable

Study these examples:

```

Sub readfromworksheet1()
    Dim fromworksheet
    ' Note that with Cells(1,2) we are referencing
    ' the first row and second column of the worksheet.
    fromworksheet = Worksheets("Sheet1").Cells(1, 2).Value
    ' The following line works just as well.
    'fromworksheet = Worksheets("Sheet1").Range("b1").Value
    MsgBox "We're in readfromworksheet1 and fromworksheet = " & _
        fromworksheet
    ' Note above, use of "_" as a continuation sign.
End Sub

```

```

Sub readfromworksheet2()
    ' Now assume we have defined a range, called testrange1,
    ' whose
    ' scope is B2:D4
    Dim fromworksheet

```



```

' Note that with Cells(1,1) we are referencing
' the first row and first column of the named range.
fromworksheet = Range("testrange1").Cells(1, 1).Value
' The following line works just as well.
'fromworksheet = Worksheets("Sheet1").Range("b1").Value
MsgBox "We're in readfromworksheet2. fromworksheet = " & _
    fromworksheet
' Note above, use of "_" as a continuation sign.
End Sub

```

A.3.5 Writing from an Excel Visual Basic Variable to a Worksheet

Just switch from left to right, e.g.,

```
Worksheets("Sheet1").Cells(1, 2).Value = fromworksheet
```

The equal sign, =, in this context is an assignment statement. It puts the stuff on the right into the stuff on the left.

A.4 Boolean Operators

Often we have to test for the truth or falsity of an expression, for example

```
MyVar > 7.3
```

will be true if MyVar has a value that is greater than 7.3. If its value is less than 7.3 the expression will be false. Note: If MyVar is Null, then the expression evaluates to Null. See comparison operators. This greatly complicates things and in these notes, I'll ignore the question of nulls.

So, expressions may be either true or false, in which case we say they have truth values. Expressions having truth values may be combined using Boolean operators to yield larger expressions, which also have truth values. The Boolean operators available in VB are: And, Or, and Not.

Each of these operators has a characteristic truth table, as follows.

expression1	expression2	(expression1 And expression2)
T	T	T
T	F	F
F	T	F
F	F	F

Table A.1: Truth Table for And

expression1	expression2	(expression1 Or expression2)
T	T	T
T	F	T
F	T	T
F	F	F

Table A.2: Truth Table for Or

expression	(Not expression)
T	F
F	T

Table A.3: Truth Table for Not

Interestingly, many other Boolean (truth-functional) operators are possible. That is, there are a lot more other truth tables possible. But, these three suffice in that with them any other possible Boolean (truth functional) operator may be defined. (How would you prove this?) In fact, Not and And are sufficient in this way, as are Not and Or. Here's something of a proof.

exp1	exp2	(exp1 And exp2)	Not(Not exp1 Or Not exp2)
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

Table A.4: Truth Table Showing Definition of And in terms of Not and Or

Can you think of a single Boolean operator that is by itself sufficient?

So, we often need Boolean combinations of statements (or expressions) in programming. The bottom line is that And, Or, and Not are sufficient for expressing anything we can possibly express in this way.

A.5 Control Structures

There are several of these in Visual Basic, and we'll look at a few of them. (And you should search the online help under "control structures.") We have already seen one, the For...Next statement.

A.5.1 For...Next

We've already seen this in action (above). The general structure for a For...Next statement is:

```
For <counter> = <start> To <end> [Step <increment>]
[statements]
Next [<counter>]
```

Note: Items in square brackets, [...], are optional. Items capitalized are required parts of the statement. Items between left and right angle brackets, <...>, are required to be filled in by the programmer. Thus, valid examples for the For...Next statement include the following.

```
For I = 1 To 3
MsgBox "Hello, world!"
Next
```

Better style is to do this:

```

For I = 1 To 3
  MsgBox "Hello, world!"
Next I

```

Or you can count down, if, e.g., `MyIncrement` is negative.

```

For MyCounter = MyStart To MyFinish Step MyIncrement
  MsgBox "MyCounter = " & MyCounter
Next MyCounter

```

Note: Be sure all these variables have reasonable values set for them before executing this statement.

A.5.2 If...Then...

This is a very useful statement in programming languages. The basic structure in VB is:

```

If <condition> Then
  [statements]
End If

```

When an `If...Then...` statement is executed, the `<condition>` is tested as a Boolean expression. If it evaluates to `True`, then the `[statements]` are executed; otherwise they are skipped and processing continues with the next statement, if any.

Note: The `<condition>` can also be an expression that returns a numeric value. If when evaluated it returns 0, that is treated as `False`. Anything else is treated as `True`.

Example:

```

If Age >= 65 Then
  NumberOfDeductions = NumberOfDeductions + 1
End If

```

Note: The `<condition>` expression may be complex. It may be an arbitrarily complex Boolean combination of statements.

A.5.3 If...Then...Else

Probably used even more often than If...Then...

```
If <condition> Then
    [statements to execute if <condition> is true]
Else
    [statement to execute if <condition> is false]
End If
```

You use If...Then...Else when you want to do one thing if a condition obtains, and another if it does not obtain. The =If(...) function in Excel is an If...Then...Else type of construct. Example: If the value in a certain cell (or variable) is valid, then display an OK message; otherwise display a not OK message.

A.5.4 Select Case

More general than If...Then...Else is Select Case.

```
Select Case <test expression>
    Case <first expression list>
        [first statements]
    Case <second expression list>
        [second statements]...

    Case Else
        [else statements]
End Select
```

Here's an example from the SuperBook:

```
Select Case TotalPoints
    Case Is < 50
        FinalGrade = "F"
    Case Is < 60
        FinalGrade = "D"
    Case Is < 70
        FinalGrade = "C"
```

```

    Case Is < 60
        FinalGrade = "B"
    Case Else
        FinalGrade = "A"
End Select

```

This runs, but there's a lot that's wrong with it. The following is much better. Why?

```

Sub testcase2()
    TotalPoints = 173
    Select Case TotalPoints
        Case 0 To 50
            FinalGrade = "F"
        Case 50 To 59
            FinalGrade = "D"
        Case 60 To 69
            FinalGrade = "C"
        Case 70 To 79
            FinalGrade = "B"
        Case 80 To 100
            FinalGrade = "A"
        Case Else
            FinalGrade = "Error in TotalPoints: " & TotalPoints
    End Select
    MsgBox "Final grade is: " & FinalGrade
End Sub

```

A.5.5 Do...Loop

There are really two forms of Do...Loop: condition-at-the-top and condition-at-the-bottom. Here they are:

```

Do {While | Until} <condition>
    [statements]
Loop

```

and

```

Do
  [statements]
Loop {While | Until} <condition>

where

{While | Until}

```

gets unpacked as either While or Until. While <condition> means so long as the condition is true, and Until <condition> means until the condition is true. The difference between the condition-at-the-top and the condition-at-the-bottom versions lies mainly in that the condition-at-the-bottom version is guaranteed to execute its [statements] at least once.

A.5.6 Exiting a Loop

Sometimes you need to break out of a loop. (Don't we all?) If you're in a For...Next structure, break out with an Exit For statement. If you're in a Do...Loop, break out with an Exit Do statement. Note: sometimes you have to do this, but it's generally considered poor programming practice. Why?

A.6 Arrays

Arrays in VB should not be confused with arrays and array commands in Excel, even though Excel's terminology invites this. All standard third-generation programming languages support arrays, and programs in these languages typically rely a lot on arrays. Arrays are rather like vectors and matrices in mathematics. A one-dimensional array is an ordered collection of values, rather like a vector, which you can access (store or retrieve values) by position. Here's a simple example.

```

' From "Code Module5" of vbtutor.xls
Sub arraytester1()
Dim I, MyFirstArray(1 To 6) As Integer
' Load up the array
For I = 1 To 6

```

```

        MyFirstArray(I) = I + 3
    Next I
    MsgBox "MyFirstArray(6) = " & MyFirstArray(6)
    ' Dump the array into a worksheet
    For I = 6 To 1 Step -1
        Worksheets("Sheet1").Cells(I, 6).Value = MyFirstArray(I)
    Next I
End Sub

```

Note: You declare an array in much the same way you declare any other variable. (But see `ReDim` in the online help.) All of the elements in an array must have the same data type. Of course, if the array is of type variant, this is pretty loose. (But you can't have, e.g., arrays within arrays in VB.)

Here's a more interesting example, using a two-dimensional array.

```

Sub arraytester2()
    Dim I, J As Integer
    Dim MySecondArray(1 To 10, 1 To 20) As Single
    ' Load up the array and dump, forcing
    ' type conversion from Integer to Single
    For I = 1 To 10
        For J = 1 To 20
            MySecondArray(I, J) = Sin(I + J)
            Worksheets("Sheet2").Cells(I, J).Value = MySecondArray(I, J)
        Next J
    Next I
End Sub

```

We can go on the high-dimensional arrays, but I think you get the idea. In Excel VB programs, you typically only need one- and two-dimensional (maybe three-dimensional) arrays.

A.7 Dialog Boxes in Excel

A.7.1 Creating a New Dialog Box

Begin by creating a new Dialog sheet. See figure A.2.

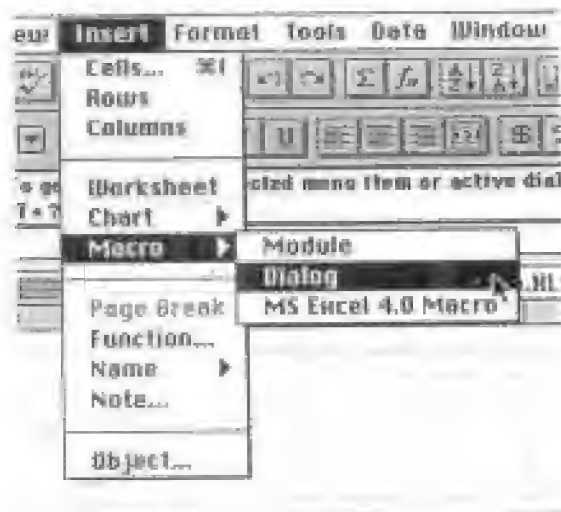


Figure A.2: Creating a New Dialog Sheet

You then get a dialog box form and the forms menu/pallet. See figure A.3.



Figure A.3: Dialog Box Form and Menu on the New Dialog Sheet

Choose List Box from the Forms pallet and draw a list box on the dialog box. See figure A.4. (Double click on selected object, the list box, or choose **Format | Object...**)

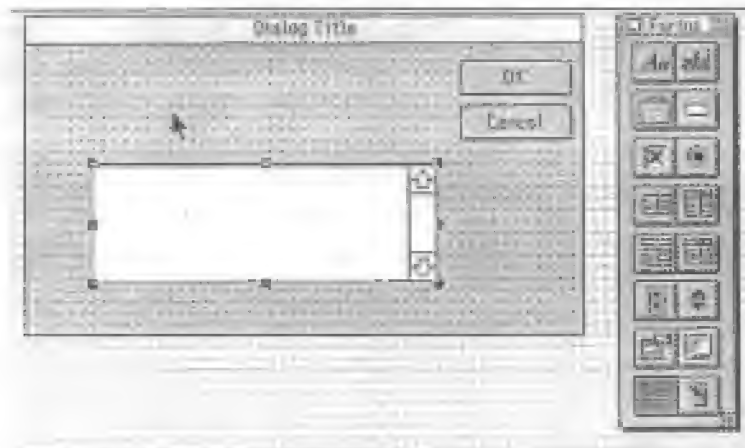


Figure A.4: Drawing a List Box

Assign links to the list box. See figure A.5.

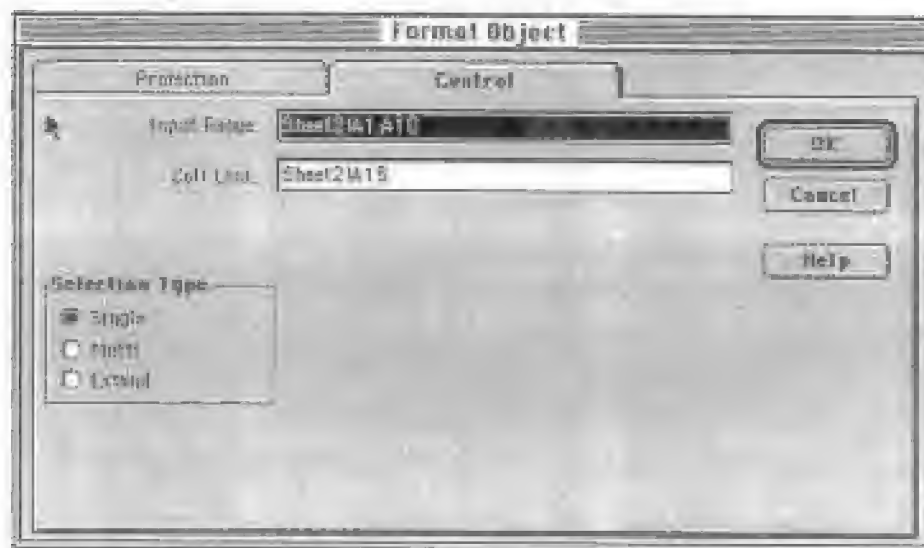


Figure A.5: Assigning Links to the List Box

Test by choosing Run Dialog from the Forms pallet. See figure A.6.

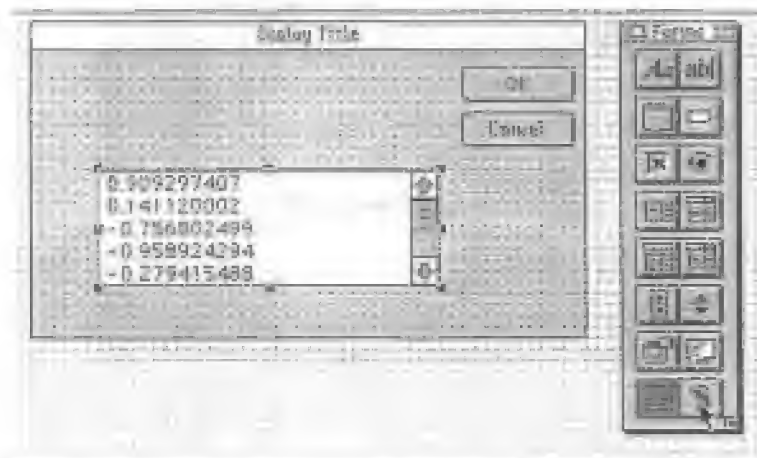


Figure A.6: Testing the List Box

View the dialog box in action and make a selection. See figure A.7.

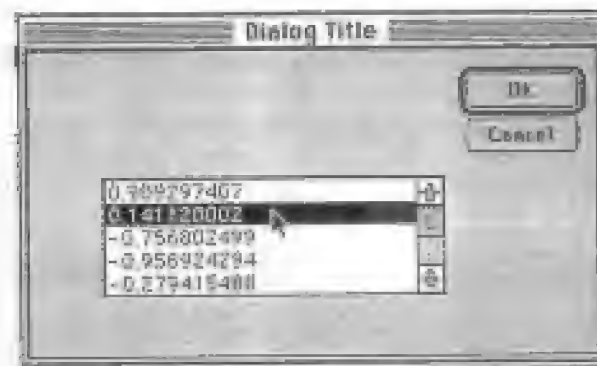


Figure A.7: The List Box in Action

Click on the OK button. Notice what is written to the Cell Link cell: the item number selected (not the value selected).

A.7.2 Calling a Dialog Box from a Sub

Now, write a subroutine (sub) that, when executed, runs and displays this dialog box. Go to a code module sheet. Try this:

```
Sub displaymyfirstdialog()
If DialogSheets("Dialog1").Show Then
    MsgBox "The user clicked the OK button."
Else
    MsgBox "The user clicked the Cancel button."
End If
End Sub
```

A.7.3 Adding a Button to a Dialog Box

Now let's see how to add a button to a dialog box. Click on the Create Button tool from the Forms pallet. Draw and edit a new button. Leave it selected. See figure A.8.



Figure A.8: Adding a New Button

Choosing Tools | Assign Macro..., assign a macro to the selected button. See Figure A.9.

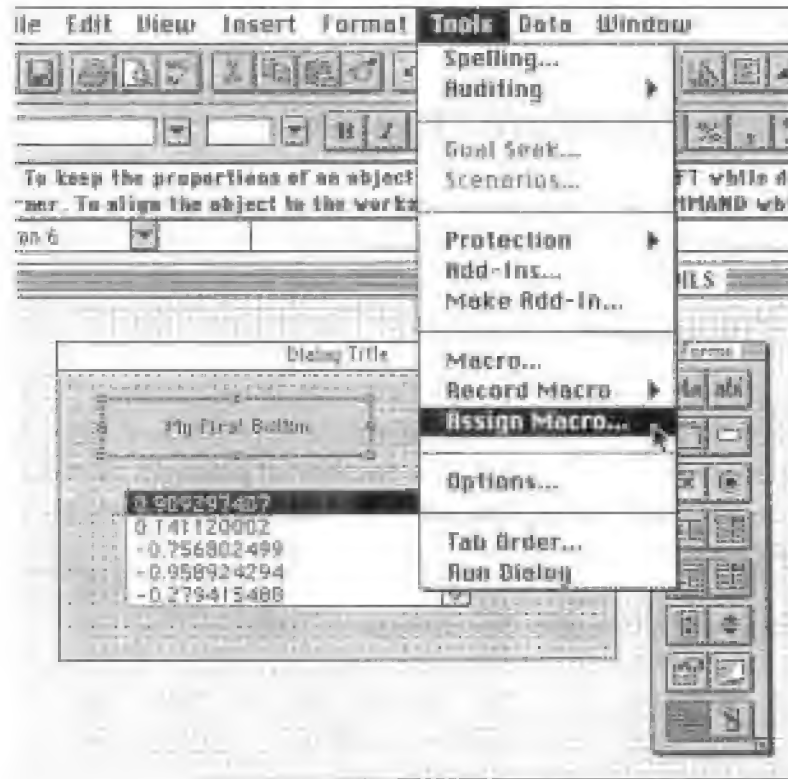


Figure A.9: Assigning a Macro to a New Button

A.7.4 Upward and Onward

This completes the very basics. You should play around and discover what else can be done.

A.8 Miscellaneous Topics

Now we'll discuss a list of useful things, things—methods and tricks—that didn't fit easily in the previous discussion.

A.8.1 Constants

Constants are like variables, except that they don't change. You use constants in order to improve the readability of your program and to help reduce errors. For example, if the maximum number of students in a classroom is 132 and you need this value a lot in your program, then you might want to consider declaring a constant. You might do this:

```
Public Const maxstudents As Integer = 132
```

Then, throughout your program, you can just use `maxstudents`, without having to worry about typing 132 or making a mistake and typing some other number. (Recall: `Option Explicit`.)

A.8.2 The Copy Method

Suppose you wish to copy one worksheet range to another worksheet range. You can do this in Excel VBA with the `copy` method. For example:

```
Sub copytest1()
    ' Suppose "carol" is the range B3:C4 on Sheet3 and
    ' "alice" is E4:F5 on Sheet3.
    ' The following works:
    Worksheets("Sheet3").Range("carol").Copy _
destination:=Worksheets("Sheet3").Range("alice")
    ' And so does this:
    Worksheets("Sheet3").Range("carol").Copy _
```

```

destination:=Worksheets("Sheet3").Cells(9, 9)
    ' and so does this:
    Worksheets("Sheet3").Range("b3:c4").Copy _
destination:=Worksheets("Sheet3").Cells(10, 2)
End Sub

```

A.8.3 Referring to Single Column or Row Ranges

Suppose the name *denise* refers to a range consisting of a single column. Then `Range("denise").Cells(1).Value` refers to the value in the topmost cell in the range.

```

Sub democells1()
    x = Range("denise").Cells(1).Value
    y = Range("denise").Cells(2).Value
    MsgBox "x = " & x & " and y = " & y
End Sub

```

A.8.4 Sorting Worksheet Ranges

See the `sort` method. In Excel VBA you can direct the sorting of a worksheet range. For example, the following subroutine sorts the range, *DaRange*, in worksheet, *DaWorkSheet*, on the column, *DaColumn*, in descending order.

```

Sub sort()
    Worksheets("DaWorkSheet").Range("DaRange").sort _
key1:=Range("DaColumn"), order1:=xlDescending
End Sub

```

A.8.5 Calling Subroutines from within Other Subroutines

A reasonable and normal thing to do. In fact it's recommended. Suppose you had a main subroutine, called *main*, and you wanted it to call three other subroutines, named *mysub1*, *mysub2*, and *mysub3*. Here's how:

```

Sub main()
    mysub1

```

```
mysub2  
mysub3  
End Sub
```

A.8.6 Calling Functions from Other Procedures

Very straightforward. See the bob-function at the start of this appendix. Then, here's an example.

```
Function bobagain(x)  
    bobagain = bob(x) * bob(x)  
End Function
```

And that's about it.

Appendix B

BasicGA: Code for Genetic Algorithms¹

B.1 Introduction

The purpose of this appendix is to lay out and discuss the code for BasicGA. BasicGA is a program having some (very) basic genetic algorithm capabilities. It was written in Visual Basic (Microsoft) and works in the Visual Basic for Applications (VBA) dialect. BasicGA works in Excel. The purpose of BasicGA is to serve as a shell or starting point for developing applications using genetic algorithms, especially in a classroom environment. My intention, thus, has been to make BasicGA as implementation-independent as possible.

Note: For program development, debugging, and other purposes, I have often substituted a stub routine in BasicGA for what would be an actual routine in a full application.² Such stubs will have "stub" appended to the actual name, sometimes prefixed and sometimes postfixed. For example, the stub for InitializeGA is InitializeGASTub but could be StubInitializeGA. Also, in this documentation, names of code objects, e.g., procedures and variables, will be given in a typewriter font, as we have just seen with InitializeGA.

¹File: dt-basicga.tex. Created: November 27, 1995, from clant-code.tex. Revised: 951222, 951128.

²A stub is a procedure that is intendedly not in final form, but is used during program development and testing.

B.2 Declarations

The purpose of this section is to describe the declared, global or public, variables for BasicGA. Here follows the declarations section of the BasicGA code. It is written in Microsoft Visual Basic 3.0 and has also been tested in the Microsoft Excel 5.0 environment.

```
' sok 951127: This is in file: GACODE.BAS in
' the folder clapepo3
' Am freezing this for now. Call it the BasicGA program.
' version 951127.
```

Option Explicit

```
'*****
'***** Below, constants declared that ***
'***** should be read in. *****

' sok 951126: Note: These program variables,
' in a non-stubbed
' environment, need to be declared in the declarations
' section. They are so declared, but I have commented
' out the declarations (see below).

'+++++
'++++ from GetGARunPara +++++

Const NumberOfGenerations = 20
    'GetNumberOfGenerations
Const PopulationSize = 100
    'GetPopulationSize
Const CrossoverRate = .77
    'GetCrossoverRate
Const MutationRate = .23
    'GetMutationRate
Const bestNSaved = 100
```

```

'GetBestNSaved
' ++++++
' ++++++ from GetModelRunPars ++++++
  Const NumberOfDecisionVariables = 4

'GetOutputSize
  Const OutputSize = 2

' ++++++

' ++++++
' ++++++ from/for InitDVarInfo/StubInitDVarInfo ++++++

Dim DecisionVariableInfo(1 To NumberOfDecisionVariables,
=> 1 To 4) As Double

Const DecisionVariableInfo11 = 5 'r, low
Const DecisionVariableInfo12 = 20 ' r, high
Const DecisionVariableInfo13 = 0 ' r, not integer
Const DecisionVariableInfo14 = 0 ' r, no grid search

Const DecisionVariableInfo21 = 10 'v, low
Const DecisionVariableInfo22 = 30 ' v, high
Const DecisionVariableInfo23 = 0 ' v, not integer
Const DecisionVariableInfo24 = 0 ' v, no grid search

Const DecisionVariableInfo31 = 15 'u, low
Const DecisionVariableInfo32 = 25 ' u, high
Const DecisionVariableInfo33 = 0 ' u, not integer
Const DecisionVariableInfo34 = 0 ' u, no grid search

Const DecisionVariableInfo41 = 200 'l, low
Const DecisionVariableInfo42 = 300 ' l, high
Const DecisionVariableInfo43 = 0 ' l, not integer
Const DecisionVariableInfo44 = 0 ' l, no grid search
' ++++++
' ++++++

```

```

'***** Above, constants declared that ***
'***** should be read in. *****

' Global variables

'++++ from GetGARunPars
' **** but explicitly declared above ++++++

    'Dim NumberOfGenerations as Integer
    'Dim PopulationSize as Integer
    'Dim CrossoverRate As Double
    'Dim MutationRate As Double
    'Dim bestNSaved as Integer

' ++++++ from GetModelRunPars
' ++++++but explicitly declared above ++++++

' Dim NumberOfDecisionVariables As Integer
' Dim OutputSize As Integer

' ++++++

Dim Index As Integer
Global CurrentGeneration() As Double
Global AbsoluteFitness() As Double
Dim ChromosomeCopySpace() As Double
Dim RelativeFitness() As Double
Dim CrossoverLikelihood() As Double
Dim BestNCurrentSaveSet() As Double
Dim LowestAbsoluteFitness As Double
Dim HighestAbsoluteFitness As Double
Dim CurrentIdNum As Double
Dim NumberOfGenerationsSoFar As Integer
Dim CrossoverPoint As Integer

Dim NoisyOutput As Integer ' 1 = show lots of output;
' 0 = don't

```


The general structure and plan for the program is simple. Everything revolves around two arrays.

First, the array `CurrentGeneration` holds the current generation of chromosomes, one chromosome per row. `CurrentGeneration` has rows running from 1 to `PopulationSize`, where `PopulationSize` is the number of individuals or chromosomes maintained in each generation. `CurrentGeneration` has columns running from 0 to `NumberOfDecisionVariables`, where `NumberOfDecisionVariables` is the number of variables at play in the model for the GA runs. Column 0 of `CurrentGeneration` holds the ID of the corresponding chromosome.

Second, the array `AbsoluteFitness` holds the results of the fitness evaluations for each chromosome in the current generation. `AbsoluteFitness` has rows running from 1 to `PopulationSize` and a row of `AbsoluteFitness` corresponds to a row of `CurrentGeneration`. `AbsoluteFitness` has columns from 1 to `OutputSize`, where `OutputSize` is the number of distinct values returned for a single chromosome by evaluation of the fitness function. Usually, `OutputSize` will equal 1, that is, only 1 value is returned: the absolute fitness of the chromosome at hand. Sometimes, however, it is useful to have the fitness function return several values. If so, then their number is indicated by `OutputSize` and it is the responsibility of the fitness function, Sub `Evaluate(I)`, to organize the response. By convention, column 1 of `AbsoluteFitness` must hold the absolute (or raw) fitness of the chromosome at hand.

`BasicGA` works by initializing `CurrentGeneration`, calculating fitnesses with `Evaluate(I)` and thereby populating `AbsoluteFitness`. Then the next generation is created. Crossingover is performed, mutation is performed, and the cycle continues until the stopping condition (a count of the generations in this code) is encountered. All this mostly happens through Sub `RunGAUntilDone`.

Now some specific comments about these declarations.

1. The following parameter is set in `InitializeGA`:
 - (a) `CurrentIDNum`. An integer, representing the ID number, or count, of a given chromosome or solution.

2. The following parameters are set in `GetCARunParams`:
 - (a) `NumberOfGenerations`. Integer, should be ≥ 0 .
 - (b) `PopulationSize`. Integer, should be ≥ 1 .
 - (c) `CrossoverRate`. Floating point, should be $\in [0, 1]$.
 - (d) `MutationRate`. Floating point, should be $\in [0, 1]$.
 - (e) `BestNSaved`. Integer, should be ≥ 0 .
3. The following parameters are set in `GetGAModelRunParams`:
 - (a) `NumberOfDecisionVariables`. Integer, should be ≥ 1 . This is the number of input variables sent to the fitness evaluation function.
 - (b) `OutputSize`. Integer, should be ≥ 1 . This is the number of output values returned by the fitness evaluation function.
4. The following parameters are set in `ReDimGAArrays`:
 - (a) `CurrentGeneration`. Declared here as nonstatic, i.e.,
`Dim CurrentGeneration() As Double`.
 - (b) `AbsoluteFitness`. Declared here as nonstatic, i.e.,
`Dim AbsoluteFitness() As Double`.
 - (c) `RelativeFitness`. Declared here as nonstatic, i.e.,
`Dim RelativeFitness() As Double`.
 - (d) `BestNCurrentSaveSet`. Declared here as nonstatic, i.e.,
`Dim BestNCurrentSaveSet() As Double`.

B.3 Sub DoTheGA: Code Structure Overview

Sub DoTheGA is the intended entry point to this program. Its structure is quite simple and the source code is given in Figure B.1.

```
' ***** Main Program *****
'
Sub DoTheGA ()

Randomize (17)
ChDir "c:\clasave\"
NoisyOutput = 1

' 1. Make preparations to run the GA.

    PrepareGA

' 2. Run the GA until the stopping condition is met

    RunGAUntilDone

' 3. Postpare the system

    PostpareGA
End Sub
```

Figure B.1: Sub DoTheGA Source Code: Main Entry Point

A few comments are in order. The purpose of Randomize (17) is to initialize the random number generator. This guarantees that on each run the same sequence of random numbers will be generated, regardless of which machine the program is run on.

ChDir "c:\clasave\"

is for the IBM PC (MS DOS) environment and will need to be changed or

commented out on the Macintosh. It assumes that a directory called `clasave` exists on the C drive. The program writes its output files to this directory.

NoisyOutput is set to 1, turning on various comments during the running of the program. Set it to 0 to turn these off.

Now, briefly, to the three subroutines called in Sub DoTheGA.

B.3.1 PrepareGA

The purpose of this subroutine is to initialize the program and to generate the first generation of chromosomes. The source code for this subroutine is given in Figure B.2.

```

Sub PrepareGA ()

' 1. Initialize the system

    InitializeGA

' 2. Validate the input data

    ValidateGAInput

' 3. Generate the initial population of chromosomes

    MakeGAGenOne

' 4. Calculate the absolute and relative
    'fitnesses for each chromosome.

    CalculateFitness

' 5. Initialize the save sets

    InitializeSaveSets
End Sub

```

Figure B.2: Sub PrepareGA Source Code

B.3.2 RunGAUntilDone

This is the subroutine that does the main work in the program. Its source code is given in Figure B.3.

```

Sub RunGAUntilDone ()
  Do Until NumberOfGenerationsSoFar >= NumberOfGenerations
    If (NoisyOutput = 1) Then
      '      MainForm.ProgressBar.Text =
==> ' "NumberOfGenerationsSoFar = " & NumberOfGenerationsSoFar
      End If
      ' Now to the main business:

      PerformCrossover
      PerformMutation
      CalculateFitness
      UpdateTheSaveSets
      SortBestNCurrentSaveSet
      NumberOfGenerationsSoFar = NumberOfGenerationsSoFar + 1
    Loop
    If (NoisyOutput = 1) Then
      '      MainForm.ProgressBar.Text =
==> ' "NumberOfGenerationsSoFar = " & NumberOfGenerationsSoFar
      End If
    End Sub
End Sub

```

Figure B.3: Sub RunGAUntilDone Source Code. Note: Lines artificially broken with my continuation symbol: ==>.

B.3.3 PostpareGA

Sub PostpareGA cleans things up once the GA has run its course. The program does two things: writes out CurrentGeneration to a file and writes out BestNCurrentSaveSet (the array holding the best N chromosomes found to this point in the GA run) to a file. The source code is given in Figure B.4.

```
Sub PostpareGA ()

' Print out final generation.
Print2FileCurGen

' Print out the best finds overall.
Print2FileBestOverall

If (NoisyOutput = 1) Then
'   MainForm.ProgressBar.Text = "All done."
End If
End Sub
```

Figure B.4: Sub PostpareGA Source Code

B.4 PrepareGA: Detailed Code Structure

B.4.1 InitializeGA

InitializeGA initializes CurrentIDNum to 0, then calls three subroutines. The first, GetGARunPars, is for obtaining information needed to make this run of the GA. The second, GetGAModelRunPars, is for obtaining particular information about the model (fitness function) that is to be applied in this particular run of the GA.

The third, ReDimGAArrays, only has the function of setting the sizes of various dynamic arrays (see declarations section, above).

1. CurrentGeneration(1 To PopulationSize,
0 to NumberOfDecisionVariables) As Double.
2. AbsoluteFitness(1 To PopulationSize,
1 To OutputSize) As Double.
3. RelativeFitness(1 To PopulationSize) As Double.
4. BestNCurrentSaveSet(1 To BestNSaved + PopulationSize,
1 To NumberOfDecisionVariables + 1 + OutputSize) As Double

GetGARunPars

The following program variables need to be initialized in this subroutine:

1. NumberOfGenerations.
2. PopulationSize.
3. CrossoverRate.
4. MutationRate.
5. BestNSaved.

GetGAModelRunPars

The following program variables need to be initialized in this subroutine:

1. NumberOfDecisionVariables.
2. OutputSize.

In addition the following array must be initialized:

1. DecisionVariableInfo.

Specifically,

```
ReDim DecisionVariableInfo(1 to _
    NumberOfDecisionVariables, 1 to 4) As Double
```

should be declared and DecisionVariableInfo initialized.

In DecisionVariableInfo each row corresponds to a decision variable. Column 1 holds the LowValue, column 2 the HighValue for the row's variable. Column 3 is 0 if the variable is not required to be an integer, and 1 otherwise. Finally, column 4 holds grid search information. (BasicGA does not have any grid search capability, but is designed to be expanded.) A 1 indicates that no grid search is being done on that variable. A number larger than 1 indicates that if a grid search is to be done, then the number represents the number of grid points to be examined for that variable. The array holds floating point numbers, and grid search counts are integers. It is up to the grid search program to make the conversion. By convention, we truncate, e.g., 3.1 stored goes to 3.

B.4.2 ValidateGAInput

The purpose of this subroutine is to validate the information collected in the InitializeGA subroutine. In the current version of the software, little or nothing is done here. Beware!

B.4.3 MakeGAGenOne

Declare ReDim CurrentGeneration(1 to PopulationSize, 0 to NumberOfDecisionVariables) As Double. Each row holds a chromosome of the current generation. Columns 1 through NumberOfDecisionVariables hold values for the corresponding decision variables. Column 0 holds the ID number of the solution.

This subroutine is very simple. It merely uses DecisionVariableInfo to load up CurrentGeneration, with the aid of a random number generator. Also, each member of the generation (i.e., each row) is given a unique ID.

B.4.4 CalculateFitness

This routine calls Evaluate(I) for each member (row) of CurrentGeneration. Evaluate(I) then calculates the fitness of that row and stores it in AbsoluteFitness. By convention, the first column of AbsoluteFitness is the absolute fitness of the corresponding row or solution. If the fitness function, Evaluate(I), returns more than one value, additional values are stored in the second, third, and so on columns of AbsoluteFitness.

Following this, CalculateRelativeFitness is called, which calculates the relative fitnesses from the absolute fitnesses and stores them in RelativeFitness, a one-dimensional array.

B.4.5 InitializeSaveSets

In the basic program, only one save set is used. BestNCurrentSaveSet stores the best N solutions so far, plus the current generation. In this subroutine, CurrentGeneration and AbsoluteFitness are read into BestNCurrentSaveSet, which is then sorted on absolute fitness in the subroutine SortBestNCurrentSaveSet.

B.5 Sub RunGAUntilDone: Detailed Code Structure

As is clear from the code for Sub RunGAUntilDone (Figure B.3 and §B.7) this procedure has five main subroutine calls. We now briefly describe each and refer the reader to the complete code listing in §B.7.

B.5.1 PerformCrossover

This is the most complex of the five subroutines, but the basic idea is simple. Using fitness proportional selection, two chromosomes are randomly drawn from CurrentGeneration. If crossover is drawn via a random number, then the two chromosomes are crossed over and the results read into the holding array, ChromosomeCopyspace. If crossover is not drawn, then the two chromosomes are simply copied into ChromosomeCopyspace. This continues until PopulationSize is reached, at which time ChromosomeCopyspace is copied back into CurrentGeneration.

B.5.2 PerformMutation

In this subroutine, the program loops through the entire array CurrentGeneration. For each entry a random number is drawn to determine whether there shall be a mutation. If there is to be a mutation, a uniform random number is drawn between the declared high and low values for the decision variable in question.

B.5.3 CalculateFitness

This routine calls the sub Evaluate which is a model-specific procedure that calculates the values for a row of the array AbsoluteFitness.

B.5.4 UpdateTheSaveSets

Only one save set is present in the program: BestNCurrentSaveSet. The program reads CurrentGeneration into columns 0 through NumberOfDecisionVariables and AbsoluteFitness into the remaining

higher-order columns, all this beginning at line `BestNSaved + 1`. This has the effect of writing over the worst rows of `BestNCurrentSaveSet`, leaving the best `BestNSaved` rows intact. The program then (next sub) sorts `BestNCurrentSaveSet` on absolute fitness.

B.5.5 SortBestNCurrentSaveSet

The program uses a simple bubble sort on column `NumberOfDecisionVariables + 1` of `BestNCurrentSaveSet`. This column is presumed to hold the absolute fitnesses of the various rows.

B.6 Sub PostpareGA: Detailed Code Structure

Sub `PostpareGA` calls two subroutines in order to write to files the current generation and the overall best `N` ($=$ `BestNSaved`) chromosomes found during the run of the GA. Source code for these two subroutines is given in Figures B.5 and B.6.

B.7 Complete Code Listing

There follows the complete listing of the code. Following the declarations section, the procedures, whether subs or functions, are in alphabetical order.

Note: For purposes of fitting the listing on the typeset page, I have occasionally broken lines. When I do this, I use the continuation symbol, `==>`, which is not part of Visual Basic.


```

Sub Print2FileBestOverall ()
Dim I, J As Integer
Dim FNameBestOverall, FNumBestOverall
Dim msg

FNumBestOverall = FreeFile
FNameBestOverall =
=> "B" & NumberOfGenerationsSoFar &
=> "F" & FNumBestOverall & ".TXT"
Open FNameBestOverall For Output As FNumBestOverall
For I = 1 To bestNSaved
    msg = ""
    For J = 0 To NumberOfDecisionVariables + OutputSize
        msg = msg & " " & BestNCurrentSaveSet(I, J)
    Next J
    Print #FNumBestOverall, msg
Next I
Close

End Sub

```

Figure B.5: Sub Print2FileBestOverall: Source Code. Note: My continuation symbol, not in Visual Basic: ==>.

```

Sub Print2FileCurGen ()
Dim I, J As Integer
Dim FNameCG, FNumCG
Dim msg

FNumCG = FreeFile
FNameCG = "C" & NumberOfGenerationsSoFar &
=> "G" & FNumCG & ".TXT"
Open FNameCG For Output As FNumCG
For I = 1 To PopulationSize
    msg = ""
    For J = 0 To NumberOfDecisionVariables
        msg = msg & " " & CurrentGeneration(I, J)
    Next J
    For J = 1 To OutputSize
        msg = msg & " " & AbsoluteFitness(I, J)
    Next J
    msg = msg & " " & RelativeFitness(I)
    Print #FNumCG, msg
Next I
Close

End Sub

```

Figure B.6: Sub Print2FileCurGen Source Code.

```
' sok 951127: This is in file: GACODE.BAS in
' the folder clapepo3
' Am freezing this for now. Call it the BasicGA program,
' version 951127.
```

Option Explicit

```
'*****
'***** Below, constants declared that ***
'***** should be read in. *****

' sok 951126: Note: These program variables,
' in a non-stubbed
' environment, need to be declared in the declarations
' section. They are so declared, but I have commented
' out the declarations (see below).

'*****
'+++ from GetGARunPars ++++++

Const NumberOfGenerations = 20
'GetNumberOfGenerations
Const PopulationSize = 100
'GetPopulationSize
Const CrossoverRate = .77
'GetCrossoverRate
Const MutationRate = .23
'GetMutationRate
Const bestNSaved = 100
'GetBestNSaved
' ++++++
' ++++++ from GetModelRunPars ++++++
Const NumberOfDecisionVariables = 4

'GetOutputSize
Const OutputSize = 2
```

```

' ++++++

' ++++++
' ++++++ from/for InitDVarInfo/StubInitDVarInfo +++++

Dim DecisionVariableInfo(1 To NumberOfDecisionVariables,
=> 1 To 4) As Double

Const DecisionVariableInfo1 = 5 'r, low
Const DecisionVariableInfo12 = 20 ' r, high
Const DecisionVariableInfo13 = 0 ' r, not integer
Const DecisionVariableInfo14 = 0 ' r, no grid search

Const DecisionVariableInfo21 = 10 'v, low
Const DecisionVariableInfo22 = 30 ' v, high
Const DecisionVariableInfo23 = 0 ' v, not integer
Const DecisionVariableInfo24 = 0 ' v, no grid search

Const DecisionVariableInfo31 = 15 'u, low
Const DecisionVariableInfo32 = 25 ' u, high
Const DecisionVariableInfo33 = 0 ' u, not integer
Const DecisionVariableInfo34 = 0 ' u, no grid search

Const DecisionVariableInfo41 = 200 'l, low
Const DecisionVariableInfo42 = 300 ' l, high
Const DecisionVariableInfo43 = 0 ' l, not integer
Const DecisionVariableInfo44 = 0 ' l, no grid search
' ++++++
' ++++++
' ++++++ Above, constants declared that ***
' ++++++ should be read in. ++++++

' Global variables

'++++ from GetGARunPara
' **** but explicitly declared above ++++++

```



```

'Dim NumberOfGenerations as Integer
'Dim PopulationSize as Integer
'Dim CrossoverRate As Double
'Dim MutationRate As Double
'Dim bestNSaved as Integer

' ++++++ from GetModelRunPars
' ++++++but explicitly declared above ++++++

'Dim NumberOfDecisionVariables As Integer
'Dim OutputSize As Integer

' ++++++

Dim Index As Integer
Global CurrentGeneration() As Double
Global AbsoluteFitness() As Double
Dim ChromosomeCopySpace() As Double
Dim RelativeFitness() As Double
Dim CrossoverLikelihood() As Double
Dim BestNCurrentSaveSet() As Double
Dim LowestAbsoluteFitness As Double
Dim HighestAbsoluteFitness As Double
Dim CurrentIdNum As Double
Dim NumberOfGenerationsSoFar As Integer
Dim CrossoverPoint As Integer

Dim NoisyOutput As Integer ' 1 = show lots of output;
' 0 = don't
Sub CalculateFitness ()
Dim I As Integer

For I = 1 To PopulationSize
    Evaluate (I)
Next I

CalculateRelativeFitness

```

```

End Sub

Sub CalculateRelativeFitness ()
    Dim I As Integer
    Dim Interval, LowestAbsoluteFitness,
    ==> HighestAbsoluteFitness As Double

    LowestAbsoluteFitness = FindLowest()
    HighestAbsoluteFitness = FindHighest()
    Interval = HighestAbsoluteFitness - LowestAbsoluteFitness
    If HighestAbsoluteFitness < LowestAbsoluteFitness Then
        MsgBox "Whoa! In CalculateRelativeFitness,
    ==> HighestAbsoluteFitness = " & HighestAbsoluteFitness & " and
    ==> LowestAbsoluteFitness = " & LowestAbsoluteFitness
    End If
    For I = 1 To PopulationSize
        If Interval > .00000001 Then
            RelativeFitness(I) = (AbsoluteFitness(I, 1) -
    ==> LowestAbsoluteFitness) / Interval
        Else
            RelativeFitness(I) = 1
        End If
    Next I

End Sub

Sub CopyStrings (String1, String2, Index)
    Dim I As Integer

    For I = 0 To NumberOfDecisionVariables
        ChromosomeCopySpace(Index, I) =
    ==> CurrentGeneration(String1, I)
        ChromosomeCopySpace(Index + 1, I) =
    ==> CurrentGeneration(String2, I)
    Next I

```

End Sub

Function Crossover () As Integer

Dim ReturnValue As Integer

If Random01Value() <= CrossoverRate Then

 ReturnValue = 1

Else

 ReturnValue = 0

End If

Crossover = ReturnValue

End Function

Sub CrossoverStrings (String1, String2, Index)

Dim I As Integer

CrossoverPoint = Int((Random01Value() *
=> (NumberOfDecisionVariables - 1)) + 1)
If CrossoverPoint >= NumberOfDecisionVariables Then
 MsgBox "Whoa! In CrossoverStrings."
End If

' Copy up to the crossover point

For I = 1 To CrossoverPoint

 ChromosomeCopySpace(Index, I) =

=> CurrentGeneration(String1, I)

 ChromosomeCopySpace(Index + 1, I) =

=> CurrentGeneration(String2, I)

Next I

' Copy past the crossover point to the end

For I = CrossoverPoint + 1 To NumberOfDecisionVariables

 ChromosomeCopySpace(Index, I) =

=> CurrentGeneration(String2, I)

 ChromosomeCopySpace(Index + 1, I) =

```

=> CurrentGeneration(String1, I)
Next I

' Assign new IDs to the chromosomes
ChromosomeCopySpace(Index, 0) = GetCurrentIDNum()
ChromosomeCopySpace(Index + 1, 0) = GetCurrentIDNum()

End Sub

' ***** Main Program *****
'
Sub DoTheGA ()

Randomize (17)
ChDir "c:\clasave\"
NoisyOutput = 1

' 1. Make preparations to run the GA.

    PrepareGA

' 2. Run the GA until the stopping condition is met

    RunGAUntilDone

' 3. Postpare the system

    PostpareGA
End Sub

Sub Evaluate (I)
' Note: This is a model-specific routine.
' And should be revised, e.g.
' p1 goes to r
Dim p1, p2, p3, p4 As Double
p1 = CurrentGeneration(I, 1)

```

```

p2 = CurrentGeneration(I, 2)
p3 = CurrentGeneration(I, 3)
p4 = CurrentGeneration(I, 4)

```

```

AbsoluteFitness(I, 1) = 2 * p1 * (1 + p2 / p3) / p4
AbsoluteFitness(I, 2) = 2 * p1 * (1 + p2 / p3) / p4

```

```

End Sub

```

```

Function FindHighest () As Double
Dim I As Integer
Dim Highest As Double
Highest = AbsoluteFitness(1, 1)
For I = 1 To PopulationSize
    If AbsoluteFitness(I, 1) > Highest Then
=> Highest = AbsoluteFitness(I, 1)
Next I
FindHighest = Highest
End Function

```

```

Function FindLowest () As Double
Dim I As Integer
Dim Lowest As Double
Lowest = AbsoluteFitness(1, 1)
For I = 1 To PopulationSize
    If AbsoluteFitness(I, 1) < Lowest Then
=> Lowest = AbsoluteFitness(I, 1)
Next I
FindLowest = Lowest
End Function

```

```

Function GetCurrentIDNum () As Double
    CurrentIdNum = CurrentIdNum + 1
    GetCurrentIDNum = CurrentIdNum

End Function

```



```

Sub GetGARunPars ()
' This is a stub right now, with the program variables to be
' initialized here declared as constants in the
' declarations section.
' But here they are:
    'Const NumberOfGenerations = 2
        'GetNumberOfGenerations
    'Const PopulationSize = 50
        'GetPopulationSize
    'Const CrossoverRate = .77
        'GetCrossoverRate
    'Const MutationRate = .23
        'GetMutationRate
    'Const BestNSaved = 50
        'GetBestNSaved
    NumberOfGenerationsSoFar = 0

```

```
End Sub
```

```

Sub GetModelRunPars ()
' This is a stub right now, with the program variables to be
' initialized here declared as constants in the
' declarations section.
' But here they are:

'NumberOfDecisionVariables = 4
    'GetNumberOfDecisionVariables
'OutputSize = 2
    'GetOutputSize

```

```

StubInitDVarInfo
    'for InitDVarInfo
End Sub

```

```
Sub InitializeGA ()
```

```
CurrentIdNum = 0
```

```

GetGARunParams
GetModelRunParams
ReDimGAArrays

End Sub

Sub InitializeSaveSets ()
Dim I, J As Integer

' Number of rows is the number in the best N save set
' plus the population size
' Number of columns is no. decision variables + ID +
' absolute fitness

' Read in CurrentGeneration array
For I = 1 To PopulationSize
    For J = 0 To NumberOfDecisionVariables
        BestNCurrentSaveSet(I, J) = CurrentGeneration(I, J)
    Next J
Next I

' Read in AbsoluteFitness array
' Note: In the BestNCurrentSaveSet array
' the absolute fitness is
' kept in column number NumberOfDecisionVariables + 1.
For I = 1 To PopulationSize
    For J = 1 To OutputSize
        BestNCurrentSaveSet(I, NumberOfDecisionVariables +
=> J) = AbsoluteFitness(I, J)
    Next J
Next I
SortBestNCurrentSaveSet
End Sub

Sub MakeGAGenOne ()
Dim I, J As Integer

```

```

Dim LowValue, HighValue As Double

For I = 1 To PopulationSize
    For J = 1 To NumberOfDecisionVariables
        LowValue = DecisionVariableInfo(J, 1)
        HighValue = DecisionVariableInfo(J, 2)
        CurrentGeneration(I, J) =
=> RandomBetween(LowValue, HighValue)
    Next J
    CurrentGeneration(I, 0) = GetCurrentIDNum()
Next I

End Sub

Sub PerformCrossover ()
    Dim I, J As Integer
    Dim String1, String2 As Integer
    Dim SumFitnesses As Double

    SumFitnesses = 0
    For I = 1 To PopulationSize
        SumFitnesses = SumFitnesses + RelativeFitness(I)
    Next I
    ' CrossoverLikelihood accumulates the probabilities of
    ' crossover. So,
    ' CrossoverLikelihood(PopulationSize) should = 1.
    CrossoverLikelihood(1) = RelativeFitness(1) / SumFitnesses
    For I = 2 To PopulationSize
        CrossoverLikelihood(I) =
=> (RelativeFitness(I) / SumFitnesses) +
=> CrossoverLikelihood(I - 1)
    Next I
    For I = 1 To PopulationSize Step 2
        String1 = RandomStrings()
        ' get a random string that can be crossed over
        String2 = RandomStrings()
        ' get a random string that can be crossed over

```

```

        If Crossover() = 1 Then
' If we do crossover here, then
            CrossoverStrings String1, String2, I
        Else ' We don't do crossover and
' we just copy the chromosomes to the next generation.
            Call CopyStrings(String1, String2, I)
        End If
    Next I

' copy back into the CurrentGeneration array
For I = 1 To PopulationSize
    For J = 0 To NumberOfDecisionVariables
        CurrentGeneration(I, J) =
=>ChromosomeCopySpace(I, J)
    Next J
Next I
End Sub

Sub PerformMutation ()
Dim I, J As Integer

For I = 1 To PopulationSize
    For J = 1 To NumberOfDecisionVariables
        If Random01Value() < MutationRate Then
            CurrentGeneration(I, J) =
=> RandomBetween(DecisionVariableInfo(J, 1),
=> DecisionVariableInfo(J, 2))
            CurrentGeneration(I, 0) = GetCurrentIDNum()
        End If
    Next J
Next I
End Sub

Sub PostpareGA ()
' Print out final generation.
Print2FileCurGen

```

```

' Print out the best finds overall.
Print2FileBestOverall

If (NoisyOutput = 1) Then
    MainForm.ProgressBar.Text = "All done."
End If

End Sub

Sub PrepareGA ()

' 1. Initialize the system
    InitializeGA

' 2. Validate the input data
    ValidateGAInput

' 3. Generate the initial population of chromosomes
    MakeGAGenQue

' 4. Calculate the absolute and relative
'     fitnesses for each chromosome.
    CalculateFitness

' 5. Initialize the save sets
    InitializeSaveSets

End Sub

Sub Print2FileBestOverall ()

```



```

Dim I, J As Integer
Dim FNameBestOverall, FNumBestOverall
Dim msg

FNumBestOverall = FreeFile
FNameBestOverall = "B" & NumberOfGenerationsSoFar &
=> "F" & FNumBestOverall & ".TXT"
Open FNameBestOverall For Output As FNumBestOverall
For I = 1 To bestNSaved
    msg = ""
    For J = 0 To NumberOfDecisionVariables + OutputSize
        msg = msg & " " & BestNCurrentSaveSet(I, J)
    Next J
    Print #FNumBestOverall, msg
Next I
Close

End Sub

Sub Print2FileCurGen ()
Dim I, J As Integer
Dim FNameCG, FNumCG
Dim msg

FNumCG = FreeFile
FNameCG = "C" & NumberOfGenerationsSoFar &
=> "G" & FNumCG & ".TXT"
Open FNameCG For Output As FNumCG
For I = 1 To PopulationSize
    msg = ""
    For J = 0 To NumberOfDecisionVariables
        msg = msg & " " & CurrentGeneration(I, J)
    Next J
    For J = 1 To OutputSize
        msg = msg & " " & AbsoluteFitness(I, J)
    Next J

```

```

        msg = msg & " " & RelativeFitness(I)
        Print #FNumCG, msg
    Next I
Close

End Sub

Function Random01Value ()
' Note: Here and only here we use the 0-1
' random number generator built into Basic.

    Random01Value = Rnd
' return a random value from the interval [0,1]

End Function

Function RandomBetween (Low, High)

    RandomBetween = (Random01Value() * (High - Low)) + Low

End Function

Function RandomStrings ()
' The purpose of this routine is to pick
' a chromosome to contribute to the next
' generation. The likelihood of being picked
' is proportional to the relative fitness of the
' chromosome
Dim I As Integer
Dim PointOnUnitInterval As Double

PointOnUnitInterval = Random01Value()
I = 1
While CrossoverLikelihood(I) < PointOnUnitInterval
    I = I + 1
Wend

```

```
RandomStrings = 1
```

```
End Function
```

```
Sub ReDimGAArrays ()
```

```
ReDim CurrentGeneration(1 To PopulationSize,  
=> 0 To NumberOfDecisionVariables) As Double  
ReDim ChromosomeCopySpace(1 To PopulationSize,  
=> 0 To NumberOfDecisionVariables) As Double  
ReDim AbsoluteFitness(1 To PopulationSize,  
=> 1 To OutputSize) As Double  
ReDim RelativeFitness(1 To PopulationSize) As Double  
ReDim BestNCurrentSaveSet(1 To bestNSaved +  
=> PopulationSize, 0 To NumberOfDecisionVariables +  
=> OutputSize) As Double  
ReDim CrossoverLikelihood(1 To PopulationSize)  
=> As Double  
End Sub
```

```
Sub RunGAUntilDone ()
```

```
Do Until NumberOfGenerationsSoFar >= NumberOfGenerations  
If (NoisyOutput = 1) Then  
MainForm.ProgressBar.Text =  
=> "NumberOfGenerationsSoFar = "  
=> & NumberOfGenerationsSoFar  
End If  
' Now to the main business:  
  
PerformCrossover  
PerformMutation  
CalculateFitness  
UpdateTheSaveSets  
SortBestNCurrentSaveSet  
NumberOfGenerationsSoFar = NumberOfGenerationsSoFar + 1  
Loop  
If (NoisyOutput = 1) Then  
MainForm.ProgressBar.Text =
```

```

=> "NumberOfGenerationsSoFar = "
=> & NumberOfGenerationsSoFar
    End If

End Sub

Sub SortBestNCurrentSaveSet ()

Dim CurrentRow, I As Integer
Dim ArraySize As Integer
Dim SortIndex As Integer
Dim NumberSwapped As Long

NumberSwapped = -1
ArraySize = bestNSaved + PopulationSize
SortIndex = NumberOfDecisionVariables + 1
    ' Above: note that in InitializeSaveSets that
    ' the absolute fitness is read
    ' into column NumberOfDecisionVariables + 1
While NumberSwapped <> 0
    NumberSwapped = 0

    For CurrentRow = 1 To ArraySize
        I = CurrentRow
        While I <= ArraySize
            If BestNCurrentSaveSet(CurrentRow, SortIndex) <
=> BestNCurrentSaveSet(I, SortIndex) Then
                SwapRows CurrentRow, I
                NumberSwapped = NumberSwapped + 1
            End If
            I = I + 1
        Wend
    Next CurrentRow
Wend

End Sub

```

```

Sub StubInitDVarInfo ()
Dim I, J As Integer

' Load up the array DecisionVariableInfo

'For I = 1 To NumberOfDecisionVariables
'    For J = 1 To 4

'        Next J
'Next I

DecisionVariableInfo(1, 1) = DecisionVariableInfo1
' r, low
DecisionVariableInfo(1, 2) = DecisionVariableInfo12
' r, high
DecisionVariableInfo(1, 3) = DecisionVariableInfo13
' r, not integer
DecisionVariableInfo(1, 4) = DecisionVariableInfo14
' r, no grid search

DecisionVariableInfo(2, 1) = DecisionVariableInfo21
' v, low
DecisionVariableInfo(2, 2) = DecisionVariableInfo22
' v, high
DecisionVariableInfo(2, 3) = DecisionVariableInfo23
' v, not integer
DecisionVariableInfo(2, 4) = DecisionVariableInfo24
' v, no grid search

DecisionVariableInfo(3, 1) = DecisionVariableInfo31
' u, low
DecisionVariableInfo(3, 2) = DecisionVariableInfo32
' u, high
DecisionVariableInfo(3, 3) = DecisionVariableInfo33
' u, not integer
DecisionVariableInfo(3, 4) = DecisionVariableInfo34
' u, no grid search

```



```

DecisionVariableInfo(4, 1) = DecisionVariableInfo41
' 1, low
DecisionVariableInfo(4, 2) = DecisionVariableInfo42
' 1, high
DecisionVariableInfo(4, 3) = DecisionVariableInfo43
' 1, not integer
DecisionVariableInfo(4, 4) = DecisionVariableInfo44
' 1, no grid search

End Sub

Sub SwapRows (Index1, Index2)

Dim I As Integer
Dim Temp() As Double
ReDim Temp(0 To NumberOfDecisionVariables +
==> OutputSize) As Double

    For I = 0 To NumberOfDecisionVariables + OutputSize
        Temp(I) = BestNCurrentSaveSet(Index1, I)
    Next I

    For I = 0 To NumberOfDecisionVariables + OutputSize
        BestNCurrentSaveSet(Index1, I) =
==> BestNCurrentSaveSet(Index2, I)
    Next I

    For I = 0 To NumberOfDecisionVariables + OutputSize
        BestNCurrentSaveSet(Index2, I) = Temp(I)
    Next I

End Sub

Sub UpdateTheSaveSets ()
Dim I, J As Integer

```

```

' Basically, this subroutine dumps the
' CurrentGeneration array
' and the AbsoluteFitness array into the
' BestNCurrentSaveSet array, by appending them after the
' current bestNSaved. Later, we sort the entire array.
' This is done as the next subroutine call in RunGAUntilDone.

' Right now only the best N overall save set
' Number of rows is the number in the save set plus
' the population size
' Number of columns is no. decision variables +
' ID + absolute fitness
For I = 1 + bestNSaved To PopulationSize + bestNSaved
    For J = 0 To NumberOfDecisionVariables
        BestNCurrentSaveSet(I, J) =
==> CurrentGeneration(I - bestNSaved, J)
    Next J
    For J = 1 To OutputSize
        BestNCurrentSaveSet(I,
==> NumberOfDecisionVariables + J) =
==> AbsoluteFitness(I - bestNSaved, J)
    Next J
Next I
End Sub

Sub ValidateGAInput ()
' Note: There's a lot more that needs doing here.

    If NumberOfGenerationsSoFar > NumberOfGenerations Then
        MsgBox "NumberOfGenerationsSoFar > NumberOfGenerations " &
==> NumberOfGenerationsSoFar & " " & NumberOfGenerations
    End If

End Sub

```


JUDGMENT IN MANAGERIAL DECISION MAKING

SECOND EDITION

MAX H. BAZERMAN

J.L. KELLOGG GRADUATE SCHOOL OF MANAGEMENT
NORTHWESTERN UNIVERSITY



JOHN WILEY & SONS
New York Chichester

Brisbane

Toronto

Singapore

TWO BIASES

The discussion of heuristics in Chapter 1 suggested that individuals develop rules of thumb to reduce the information-processing demands of decision making. These rules of thumb provide managers with efficient ways of dealing with complex problems that produce good decisions a significant proportion of the time. However, heuristics also lead managers to *systematically* biased outcomes. A cognitive bias (or simply *bias* throughout this book) refers to situations in which a heuristic is inappropriately applied by an individual in reaching a decision.

This chapter is written to provide you with the opportunity to audit your own decision making and identify the biases that affect you. A number of problems are presented that allow you to examine your problem solving and learn how your judgments compare to the judgments of others. The quiz items are then used to illustrate 13 predictable biases to which managers are prone, and that frequently lead to judgments that systematically deviate from rationality.

To start out, consider the following two problems:

Problem 1: The following 10 corporations were ranked by *Fortune* magazine to be among the 500 largest United States-based firms according to sales volume for 1987:

Group A: Gillette, Coca-Cola Enterprises, Lever Brothers, Apple Computers, Hershey Foods

Group B: Coastal, Weyerhaeuser, Northrup, CPC International, Champion International

Which group of five organizations listed (A or B) had the larger total sales volume?

Problem 2: (Adapted from Kahneman and Tversky, 1973)

The best student in my introductory MBA class this past semester writes poetry and is rather shy and small in stature. What was the student's undergraduate major:

- (A) Chinese studies or
- (B) Psychology?

WHARTON PHOTOGRAPHIC



What are your answers? If you answered A for each of the two problems, you may gain comfort in knowing that the majority of respondents choose A. If you answered B, you are part of the minority. In this case, however, the minority represents the correct response. All corporations in group B were ranked in the Fortune 100, while none of the corporations in group A had sales as large. In fact, the total sales for group B was more than double the total sales for group A. In the second problem, the student was actually a psychology major, but more important, selecting psychology as the student's major represents a more rational response given the limited information.

Problem 1 illustrates the availability heuristic discussed in Chapter 1. In this problem, group A contains consumer firms, while group B consists of industrial firms and holding companies. Most of us are more familiar with consumer firms than conglomerates and can more easily generate information in our minds about their size. If we were aware of our bias resulting from the availability heuristic, we would recognize our differential exposure to this information and adjust, or at least question, our judgments accordingly.

Problem 2 illustrates the representativeness heuristic. The reader who responds "Chinese studies" has probably overlooked relevant *base-rate* information—namely, the likely ratio of Chinese studies majors to psychology majors within the MBA student population. When asked to reconsider the problem in this context, most people change their response to "psychology" in view of the relative scarcity of Chinese studies majors seeking MBAs. This example emphasizes that logical base-rate reasoning is often overwhelmed by qualitative judgments drawn from available descriptive information.

The purpose of problems 1 and 2 is to demonstrate how easily faulty conclusions are drawn when we overrely on cognitive heuristics. In the remainder of this chapter, additional problems are presented to further increase your awareness of the impact of heuristics on your decisions and to help you develop an appreciation for the systematic errors that emanate from overdependence on them. The thirteen biases examined in this chapter are relevant to virtually all individuals. Each of the biases is related to at least one of the three judgmental heuristics introduced in Chapter 1, and an effort has been made to categorize them accordingly. However, it is important to remember that the way our minds work in developing and using heuristics is not straightforward. Often our heuristics work in tandem in approaching cognitive tasks.

The goal of the chapter is to help you "unfreeze" your decision-making patterns and realize how easily heuristics become biases when improperly applied. By working on numerous problems that demonstrate the failures of these heuristics, you will become more aware of the biases in your decision making. By learning to spot these biases, you can improve the quality of your decisions.

Before reading further, please take a few minutes to respond to the problems outlined in Table 2.1. They will be used to illustrate the 13 decision biases presented in the remainder of this chapter.

Table 2.1 Chapter Problems

Respond to the following 11 problems before reading the chapter.

Problem 3: Which is riskier:

- a. driving a car on a 400-mile trip?
- b. flying on a 400-mile commercial airline flight?

Problem 4: Are there more words in the English language

- a. that start with an *r*
- b. for which *r* is the third letter?

Problem 5: Mark is finishing his MBA at a prestigious university. He is very interested in the arts and at one time considered a career as a musician. Is Mark more likely to take a job

- a. in the management of the arts?
- b. with a management consulting firm?

Problem 6: In 1986, two research groups sampled consumers on the driving performance of the Dodge Colt versus the Plymouth Champ in a blind road test; that is, the consumers did not know when they were driving the Colt or the Champ. As you may know, these cars were identical; only the marketing varied.

One research group (A) sampled 66 consumers each day for 60 days (a large number of days to control for weather and other variables), while the other research group (B) sampled 22 consumers each day for 50 days. Which consumer group observed more days in which 60 percent or more of the consumers tested preferred the Dodge Colt:

- a. Group A?
- b. Group B?

Problem 7: You are about to hire a new central-region sales director for the fifth time this year. You predict that the next director should work out reasonably well, since the last four were "lemons," and the odds favor hiring at least one good sales director in five tries. This thinking is

- a. Correct.
- b. Incorrect.

Problem 8: You are the sales forecaster for a department store chain with nine locations. The chain depends on you for quality projections of future sales in order to make decisions on staffing, advertising, information system developments, purchasing, renovation, and the like. All stores are similar in size and merchandise selection. The main difference in their sales occurs because of location and random fluctuations. Sales for 1989 were as follows:

Store	1989	1991
1	\$12,000,000	\$ _____
2	11,500,000	_____
3	11,000,000	_____
4	10,500,000	_____

Table 2.1 (Continued)

5	10,000,000	_____
6	9,500,000	_____
7	9,000,000	_____
8	8,500,000	_____
9	8,000,000	_____
TOTAL	\$90,000,000	\$99,000,000

Your economic forecasting service has convinced you that the best estimate of total sales increases between 1989 and 1991 is 10 percent (to \$99,000,000). Your task is to predict 1991 sales for each store. Since your manager believes strongly in the economic forecasting service, it is imperative that your total sales equal \$99,000,000.

Problem 9: Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and she participated in antinuclear demonstrations.

Rank order the following eight descriptions in terms of the probability (likelihood) that they describe Linda:

- _____ a. Linda is a teacher in an elementary school.
- _____ b. Linda works in a bookstore and takes yoga classes.
- _____ c. Linda is active in the feminist movement.
- _____ d. Linda is a psychiatric social worker.
- _____ e. Linda is a member of the League of Women Voters.
- _____ f. Linda is a bank teller.
- _____ g. Linda is an insurance salesperson.
- _____ h. Linda is a bank teller who is active in the feminist movement.

Problem 10: A newly hired engineer for a computer firm in the Boston metropolitan area has four years of experience and good all-around qualifications. When asked to estimate the starting salary for this employee, my secretary (knowing very little about the profession or the industry) guessed an annual salary of \$23,000. What is your estimate?

\$ _____ per year.

Problem 11: Which of the following appears most likely?
Which appears second most likely?

- a. Drawing a red marble from a bag containing 50 percent red marbles and 50 percent white marbles.
- b. Drawing a red marble seven times in succession, with replacement (a selected marble is put back in the bag before the next marble is selected), from a bag containing 90 percent red marbles and 10 percent white marbles.
- c. Drawing at least one red marble in seven tries, with replacement, from a bag containing 10 percent red marbles and 90 percent white marbles.

Problem 12: Listed below are 10 uncertain quantities. Do not look up any information on these items. For each, write down your best estimate of the quantity. Next, put a lower

Table 2.1 (Continued)

and upper bound around your estimate, such that you are 98 percent confident that your range surrounds the actual quantity.

- _____ a. Mobil Oil's sales in 1987
- _____ b. IBM's assets in 1987
- _____ c. Chrysler's profit in 1987
- _____ d. The number of U.S. industrial firms in 1987 with sales greater than those of Consolidated Papers
- _____ e. The U.S. gross national product in 1945
- _____ f. The amount of taxes collected by the U.S. Internal Revenue Service in 1970
- _____ g. The length (in feet) of the Chesapeake Bay Bridge-Tunnel
- _____ h. The area (in square miles) of Brazil
- _____ i. The size of the black population of San Francisco in 1970
- _____ j. The dollar value of Canadian exports of lumber in 1977

Problem 13: (Adapted from Einhorn and Hogarth, 1978)

It is claimed that when a particular analyst predicts a rise in the market, the market always rises. You are to check this claim. Examine the information available about the following four events (cards):

Card 1
Prediction:
Favorable
report

Card 2
Prediction:
Unfavorable
report

Card 3
Outcome:
Rise in the
market

Card 4
Outcome:
Fall in the
market

You currently see the predictions (cards 1 and 2) or outcomes (cards 3 and 4) associated with four events. You are seeing one side of a card. On the other side of cards 1 and 2 is the actual outcome, while on the other side of cards 3 and 4 is the prediction that the analyst made. Evidence about the claim is potentially available by turning over the card(s). Which cards would you turn over for the evidence that you need to check the analyst's claim? (Circle the appropriate cards.)

BIASES EMANATING FROM THE AVAILABILITY HEURISTIC

Bias 1—Ease of Recall (based upon vividness and recency)

Problem 3: Which is riskier:

- a. driving a car on a 400-mile trip?
- b. flying on a 400-mile commercial airline flight?

Many people respond that flying in a commercial airliner is far riskier than driving a car. The media's tendency to sensationalize airplane crashes contributes to this perception. In actuality, the safety record for flying is far better than that for driving. Thus, this example demonstrates that a particularly *vivid* event will systematically influence the probability assigned to that type of event by an individual in the future. This bias occurs because vivid events are more easily remembered and consequently are more available when making judgments.

Consider another example. A buyer of women's wear for a leading department store is assessing her purchasing needs in footwear. To fill the demand for casual shoes, she needs to choose between a proven best-selling brand of running shoes and a newer line of boating shoes. The buyer recalls having seen a number of friends wearing boating shoes at a recent party and concludes that demand for boating shoes is increasing. She decides to order more boating shoes and reduce her order of the historically popular running shoes.

In making this choice, the buyer has biased her ordering decision based upon limited data and the ease with which it came to mind. The buyer judged the demand for boating shoes by the availability of her recollection of a recent party. Under the influence of this bias, she will be consistently less likely to buy popular shoes worn by other groups with whom she tends not to socialize—even though aggregate demand for these alternative styles may be higher.

Tversky and Kahneman (1974) argue that when an individual judges the frequency of an event by the *availability* of its instances, an event whose instances are more easily recalled will appear more numerous than an event of equal frequency whose instances are less easily recalled. They cite evidence of this bias in a lab study in which individuals were read lists of names of well-known personalities of both sexes and asked to determine whether the lists contained the names of more men or women. Different lists were presented to two groups. One group received lists bearing the names of women who were relatively more famous than the listed men, but included more men's names overall. The other group received lists bearing the names of men who were relatively more famous than the listed women, but included more women's names overall. In each case, the subjects incorrectly guessed that the sex that had the more famous personalities was the more numerous.

Many examples of this bias can be observed in the decisions made by managers in the workplace. The following came from the experience of one of my MBA students: As a purchasing agent, he had to select one of several possible suppliers. He chose the firm whose name was the most familiar to him. He later found out that the salience of the name resulted from recent adverse publicity concerning the firm's extortion of funds from client companies!

Managers conducting performance appraisals often fall victim to the availability heuristic. Working from memory, the vivid instances relating to an employee that are more easily recalled from memory (either pro or con) will appear more numerous and will therefore be weighted more heavily in the performance appraisal. Managers also give more weight to performance during the three months prior to the evaluation than to the previous nine months of the evaluation period.

Many consumers are annoyed by repeated exposure to the same advertising message and often wonder why the advertiser doesn't give more useful information, without repeating it so many times. After all, we are smart enough to understand it the first time! Unfortunately, both the frequency and the vividness of the message have been shown to affect our purchasing. This bombardment of repeated, uninformative messages makes the product more easily recalled from memory and is often the best way to get us to buy a product (Alba and Marmorstein, 1987).

Because of our susceptibility to vividness and recency, Kahneman and Tversky suggest that we are particularly prone to overestimating unlikely events. For instance, if we actually witness a burning house, the impact on our assessment of the probability of such accidents is probably greater than the impact of reading about a fire in the local newspaper. The direct observation of such an event makes it more salient to us. Similarly, Slovic and Fischhoff (1977) discuss the implications of the misuse of the availability heuristic on the perceived risks of nuclear power. They point out that any discussion of the potential hazards, regardless of likelihood, will increase the memorability of those hazards and increase their perceived risks.

The stock market provides some telling examples of the tendency to overreact to vivid and recent information in this way. After the April 1986 nuclear accident at Chernobyl in the Soviet Union, U.S. investors sold their nuclear stocks, which caused a dramatic fall in prices. Yet the real safety of the nuclear systems did not change dramatically as a result of the Chernobyl accident. Similarly, the stock of Union Carbide fell 30 percent within three weeks of the December 1984 tragedy at its chemical plant in Bhopal, India. Few investors stopped to realize that Union Carbide might reach an acceptable out-of-court settlement. It was more salient to imagine Union Carbide being hit with a devastating financial penalty. More rational investors who bought the stock at its low point turned a hefty profit—even before the stock moved up higher on an unsuccessful takeover bid (Curran, 1987).

Bias 2—Retrievability (based upon memory structures)

Problem 4: Are there more words in the English language

- a. that start with an *r*?
- b. for which *r* is the third letter?

If you responded "start with an *r*," you have joined the majority. Unfortunately, this is again the incorrect answer. Kahneman and Tversky (1973) explain that people typically solve this problem by first recalling words that begin with *r* (like *ran*) and words that have an *r* as the third letter (like *bar*). The relative difficulty of generating words in each of these two categories is then assessed. If we think of our mind as being organized like a dictionary, it is easier to find lots of words that start with an *r*. The dictionary, and our minds, are less efficient at

finding words that follow a rule that is inconsistent with the organizing structure—like words that have an *r* as the third letter. Thus, words that start with a particular letter are more available from memory, even though most consonants are more common in the third position than in the first.

Just as our tendency to alphabetize affects our vocabulary-search behavior, organizational modes affect information-search behavior within our work lives. We structure organizations to provide order, but this same structure can lead to confusion if the presumed order is not exactly as suggested. For example, many organizations have a management information systems (MIS) division that has generalized expertise in computer applications. Assume that you are a manager in a product division and need computer expertise. If that expertise exists within MIS, the organizational hierarchy will lead you to the correct resource. If they lack the expertise in a specific application, but it exists elsewhere in the organization, the hierarchy is likely to bias the effectiveness of your search. I am not arguing for the overthrow of organizational hierarchies; I am merely identifying the dysfunctional role of hierarchies in potentially biasing search behavior. If we are aware of the potential bias, we need not be affected by this limitation.

Retail store location is influenced by the way in which consumers search their minds when seeking a particular commodity. Why are multiple gas stations at the same intersection? Why do "upscale" retailers want to be in the same mall? Why are the best bookstores in a city often all located within a couple blocks of each other? An important reason for this pattern is that consumers learn the "location" for a particular type of product or store and organize their minds accordingly. To maximize traffic, the retailer needs to be in the location that consumers associate with this type of product or store.

Bias 3—Presumed Associations

People frequently fall victim to the availability bias in their assessment of the likelihood of two events occurring together. For example, consider the following questions: Is marijuana use related to delinquency? Are couples who get married under the age of 25 more likely to have bigger families? How would you respond if asked these questions? In assessing the marijuana question, most people typically remember several delinquent marijuana users and assume a correlation or not based upon the availability of this mental data. However, proper analysis would include recalling four groups of observations: marijuana users who are delinquents, marijuana users who are not delinquents, delinquents who do not use marijuana, and nondelinquents who do not use marijuana. The same analysis applies to the marriage question. Proper analysis would include four groups: couples who married young and have large families, couples who married young and have small families, couples who married older and have large families, and couples who married older and have small families. Indeed, there are always at least four separate situations to be considered in assessing the association between two dichotomous events, but our everyday decision making commonly ignores this scientifically valid fact.

Chapman and Chapman (1967) have noted that when the probability of two events co-occurring is judged by the availability of perceived co-occurring instances in our minds, we usually assign an inappropriately high probability that the two events will co-occur again. Thus, if we know a lot of marijuana users who are delinquents, we assume that marijuana use is related to delinquency. Similarly, if we know of a lot of couples who married young and have had large families, we assume that this trend is more prevalent than it may actually be. In testing for this bias, Chapman and Chapman provided subjects with information about hypothetical psychiatric patients. The information included a written clinical diagnosis of the "patient" and a drawing of a person made by the "patient." The subjects were asked to estimate the frequency with which each diagnosis (for example, suspiciousness or paranoia) was accompanied by various facial and body features in the drawings (for example, peculiar eyes). Throughout the study, subjects markedly overestimated the frequency of pairs commonly associated together by social lore. For example, diagnoses of suspiciousness were overwhelmingly associated with peculiar eyes. In addition, Chapman and Chapman found that conclusions, such as the just noted, were extremely resistant to change, even in the face of contradictory information. Furthermore, the overwhelming impact of this bias toward presumed associations prevented the subjects from detecting other relationships that were, in fact, present.

Summary A lifetime of experience has led us to believe that, in general, more frequent events are recalled in our minds more easily than less frequent ones, and likely events are easier to recall than unlikely events. In response to this learning, we have developed the availability heuristic for estimating the likelihood of events. In many instances, this simplifying heuristic leads to accurate, efficient judgments. However, as these first three biases (ease of recall, retrievability, and presumed associations) indicate, the misuse of the availability heuristic can lead to systematic errors in managerial judgment. We too easily assume that our available recollections are truly representative of some larger pool of occurrences that exists outside our range of experience.

BIASES EMANATING FROM THE REPRESENTATIVENESS HEURISTIC

Bias 4—Insensitivity to Base Rates

Problem 5: Mark is finishing his MBA at a prestigious university. He is very interested in the arts and at one time considered a career as a musician. Is Mark more likely to take a job

- a. in the management of the arts?
- b. with a management consulting firm?

How did you decide on your answer? How do most people make this assessment? How *should* people make this assessment? Using the representa-

tiveness heuristic discussed in Chapter 1, most people approach this problem by analyzing the degree to which Mark is representative of their image of individuals who take jobs in each of the two areas. Consequently, they usually conclude "in the management of the arts." However, as we discussed in the first part of this chapter, this response overlooks relevant base-rate information. Reconsider the problem in light of the fact that a much larger number of MBAs take jobs in management consulting than in the management of the arts—relevant information that should enter into any reasonable prediction of Mark's career path. With this base-rate data, it is only reasonable to predict "management consulting."

Judgmental biases of this type frequently occur when individuals cognitively ask the wrong question. If you answered "in the management of the arts," you were probably thinking in terms of the question "How likely is it that a person working in the management of the arts would fit Mark's description?" However, the problem necessitates the question "How likely is it that someone fitting Mark's description will choose arts management?" By itself, the representativeness heuristic incorrectly leads to a similar answer to both questions, since this heuristic leads individuals to compare the resemblance of the personal description and the career path. However, when base-rate data is considered, it is irrelevant to the first question listed, but it is crucial to a reasonable prediction on the second question. While a large percentage of individuals in arts management may fit Mark's description, there are undoubtedly a larger absolute number of management consultants fitting Mark's description because of the relative preponderance of MBAs in management consulting.

An interesting finding of the research done by Kahneman and Tversky (1972, 1973) is that subjects do use base-rate data correctly when no other information is provided. For example, in the absence of a personal description of Mark in Problem 5, people will choose "management consulting" based on the past frequency of this career path for MBAs. Thus, people understand the relevance of base-rate information, but tend to disregard this data when descriptive data is also available.

Bias 5—Insensitivity to Sample Size

Problem 6: In 1986, two research groups sampled consumers on the driving performance of the Dodge Colt versus the Plymouth Champ in a blind road test; that is, the consumers did not know when they were driving the Colt or the Champ. As you may know, these cars were identical; only the marketing varied.

One research group (A) sampled 66 consumers each day for 60 days (a large number of days to control for weather and other variables), while the other research group (B) sampled 22 consumers each day for 50 days. Which consumer group observed more days in which 60 percent or more of the consumers tested preferred the Dodge Colt:

- a. group A?
- b. group B?

Most individuals expect research group A to provide more 60-percent days for the Dodge Colt, because of the larger number of sample days—in other words, there are 60 chances compared to 50. In contrast, simple statistics tells us that it is much more likely to observe more 60-percent days on daily samples of 22 than on daily samples of 66, and the correct answer is group B. This is because a large sample is far less likely to stray from the expected 50-percent preference split between the Dodge Colt and Plymouth Champ—since the cars are identical. (The interested reader can verify this fact with the use of an introductory statistics book.)

While the importance of sample size is fundamental in statistics, Kahneman and Tversky (1974) note that it "is evidently not part of people's repertoire of intuitions" (p. 1126). Why is this? When responding to problems dealing with sampling, people often use the representativeness heuristic. In their minds, they ask the question, Which group is likely to have more days in which the results are skewed to 60 percent for the Dodge Colt instead of the expected 50 percent? From there, the representative heuristic leads them to focus on the number of days as the pertinent variable for comparison. They then conclude that the group covering the greater number of total days will experience the greater number of total deviations. However, this analogy ignores the issue of sample size—which is critical to an accurate assessment of the problem.

Tversky and Kahneman (1974) first discovered this bias toward ignoring the role of sample size, even when these data were emphasized in the formation of the problem, in testing the following research problem:

A certain town is served by two hospitals. In the larger hospital about 45 babies are born each day, and in the smaller hospital about 15 babies are born each day. As you know, about 50 percent of all babies are boys. However, the exact percentage varies from day to day. Sometimes it may be higher than 50 percent, sometimes lower. For a period of one year, each hospital recorded the days on which more than 60 percent of the babies born were boys. Which hospital do you think recorded more such days?

The larger hospital? (21)

The smaller hospital? (21)

About the same? (53)

(that is, within 5 percent of each other)

The values in parentheses represent the number of individuals who chose each answer. As explained earlier, sampling theory tells us that the expected number of days on which more than 60 percent of the babies are boys is much greater in the small hospital, since a large sample is less likely to stray from the mean. However, most subjects judged the probability to be the same in each hospital, effectively ignoring sample size.

Consider the implications of this bias in advertising, where people trained in market research understand the need for a sizable sample, but employ this bias to the advantage of their clients. "Four out of five dentists surveyed recommend sugarless gum for their patients who chew gum." There is no mention of the number of dentists involved in the survey and the fact that without these data, the results of the survey are meaningless. If only 5 or 15 dentists were surveyed, the size of the sample would not be generalizable to the overall population of dentists.

Bias 6—Misconceptions of Chance

Problem 7: You are about to hire a new central-region sales director for the fifth time this year. You predict that the next director should work out reasonably well, since the last four were "lemons," and the odds favor hiring at least one good sales director in five tries. This thinking is

- a. correct.
- b. incorrect.

Most people are comfortable with the foregoing logic, or at least have been guilty of using similar logic in the past. However, the performance of the first four sales directors will not directly affect the performance of the fifth sales director, and the logic in problem 7 is incorrect. Most individuals frequently rely upon their intuition and the representativeness heuristic and incorrectly conclude that a poor performance is unlikely because the probability of getting five "lemons" in a row is extremely low. Unfortunately, this logic ignores the fact that we have already witnessed four "lemons" (an unlikely occurrence), and the performance of the fifth sales director is independent of that of the first four.

This question parallels Kahneman and Tversky's (1972) work in which they show that people expect that a sequence of random events will "look" random. They present evidence of this bias in their finding that subjects routinely judged the sequence of coin flips H-T-H-T-T-H to be more likely than H-H-H-T-T-T, which does not "appear" random, and more likely than the sequence H-H-H-H-T-H, which does not represent the equal likelihood of heads and tails. Simple statistics, of course, tell us that each of these sequences is equally likely because of the independence of multiple random events.

Problem 7 moves beyond dealing with random events in recognizing our inappropriate tendency to assume that random and nonrandom events will "balance out." Will the fifth sales director work out well? Maybe. You might spend more time and money on selection, and the randomness of the hiring process may favor you this time. But your earlier failures in hiring sales directors will not directly affect the performance of the new sales director.

The logic concerning misconceptions of chance provides a process expla-

nation of the gambler's fallacy. After holding bad cards on ten hands of poker, the poker player believes that he is due for a good hand. After winning \$1,000 in the Pennsylvania State Lottery, a woman changes her regular number—because after all, how likely is it that the same number will come up twice? Tversky and Kahneman (1974) note that "Chance is commonly viewed as a self-correcting process in which a deviation in one direction induces a deviation in the opposite direction to restore the equilibrium. In fact, deviations are not corrected as a chance process unfolds, they are merely diluted."

In each of the preceding examples, individuals expected probabilities to even out. In some situations, our minds misconceptualize chance in exactly the opposite way. In sports (basketball specifically), we often think of a particular player as having a "hot hand" or "being on a good streak." If your favorite player has hit his last four shots, is the probability of his making his next shot higher, lower, or the same as the probability of his making a shot without the preceding four hits? Most sports fans, sports commentators, and players believe that the answer is "higher." In fact, there are many biological, emotional, and physical reasons that this answer could be correct. However, it is wrong! Gilovich, Vallone, and Tversky (1985) did an extensive analysis of the shooting of Philadelphia 76ers and Boston Celtics and found that immediately prior shot performance did not change the likelihood of success on the upcoming shot. Out of all of the findings in this book, this is the effect that my managerial students have had the hardest time believing. The reason is that we can all remember sequences of five hits in a row: streaks are part of our conception of chance in athletic competition. However, our minds do not categorize a string of "four in a row" as being a situation in which "he missed his fifth shot." As a result, we have a misconception of connectedness, when, in fact, chance (or the player's normal probability of success) is really in effect.

The belief in the hot hand is especially interesting because of its implication for how players play the game. Passing the ball to the player who is "hot" is commonly endorsed as a good strategy. It can also be expected that the opposing team will concentrate on guarding the hot player. Another player, who is less "hot" but is equally skilled, may have a better chance of scoring. Thus the belief in the "hot hand" is not just erroneous, but could also be costly if you play professional basketball.

Tversky and Kahneman's (1971) work shows that misconceptions of chance are not limited to gamblers, sportsfans, or laypersons. Research psychologists also fall victim to the "law of small numbers." They believe that sample events should be far more representative of the population from which they were drawn than simple statistics would dictate. The researchers put too much faith in the results of initial samples and grossly overestimate the replicability of empirical findings. This suggests that the representativeness heuristic may be so well institutionalized in our decision processes that even scientific training and its emphasis on the proper use of statistics may not effectively eliminate its biasing influence.

Bias 7—Regression to the Mean

Problem 8: You are the sales forecaster for a department store chain with nine locations. The chain depends on you for quality projections of future sales in order to make decisions on staffing, advertising, information system developments, purchasing, renovation, and the like. All stores are similar in size and merchandise selection. The main difference in their sales occurs because of location and random fluctuations. Sales for 1989 were as follows:

Store	1989	1991
1	\$12,000,000	\$_____
2	11,500,000	_____
3	11,000,000	_____
4	10,500,000	_____
5	10,000,000	_____
6	9,500,000	_____
7	9,000,000	_____
8	8,500,000	_____
9	8,000,000	_____
TOTAL	\$90,000,000	\$99,000,000

Your economic forecasting service has convinced you that the best estimate of total sales increases between 1989 and 1991 is 10 percent (to \$99,000,000). Your task is to predict 1991 sales for each store. Since your manager believes strongly in the economic forecasting service, it is imperative that your total sales are equal to \$99,000,000.

Think about the processes used to answer this problem. Consider the following logical pattern of thought: "The overall increase in sales is predicted to be 10 percent ($\$99,000,000 - \$90,000,000 / \$90,000,000$). Lacking any other specific information on the stores, it makes sense to simply add 10 percent to each 1989 sales figure to predict 1991 sales. This means that I predict sales of \$13,200,000 for store 1, sales of \$12,650,000 for store 2, and so on." This logic, in fact, is the most common approach in responding to this item. Unfortunately, this logic is faulty.

Why was the logic presented faulty? Statistical analysis would dictate that we first assess the predicted relationship between 1989 and 1991 sales. This relationship, formally known as a **correlation**, can vary from total independence (that is, 1989 sales do not predict 1991 sales) to perfect correlation (1989 sales are a perfect predictor of 1991 sales). In the former case, the lack of a relationship between 1989 and 1991 sales would mean that 1989 sales would provide absolutely no information about 1991 sales, and your best estimates of 1991 sales would be equal to total sales divided by the number of stores ($\$99,000,000$ divided by 9 equals \$11,000,000). However, in the latter case of perfect predictability between 1989 and 1991 sales, our initial logic of

simply extrapolating from 1989 performance by adding 10 percent to each store's performance would be completely accurate. Obviously, 1989 sales are most likely to be *partially predictive* of 1991 sales—falling somewhere between independence and perfect correlation. Thus, the best prediction for store 1 should lie between \$11,000,000 and \$13,200,000, depending upon how predictive you think 1989 sales will be of 1991 sales. The key point is that in virtually all such predictions, you should expect the naive \$13,200,000 estimate to regress toward the overall mean (\$11,000,000).

In a study of sales forecasting, Cox and Summers (1987) examined the judgments of professional retail buyers. They examined the sales data from 2 department stores for 6 different apparel styles for a total of 12 different sales forecasts over a 2-week period. They found that sales between the 2 weeks regressed to the mean. However, the judgment of all 31 buyers from 5 different department stores failed to reflect the tendency for regression to the mean. As a result, Cox and Summers argued that a sales-forecasting model that considered regression to the mean could outperform the judgments of all 31 professional buyers.

Many effects regress to the mean. Brilliant students frequently have less successful siblings. Short parents tend to have taller children. Great rookies have mediocre second years (the "sophomore jinx"). Firms that have outstanding profits one year tend to have lesser performances the next year. In each case, individuals are often surprised when made aware of these predictable patterns of regression to the mean.

Why is the regression-to-the-mean concept, while statistically valid, counterintuitive? Kahneman and Tversky (1973) suggest that the representativeness heuristic accounts for this systematic bias in judgment. They argue that individuals typically assume that future outcomes (for example, 1991 sales) will be maximally representative of past outcomes (1989 sales). Thus, we tend to naively develop predictions that are based upon the assumption of perfect correlation with past data.

In some unusual situations, individuals do intuitively expect a regression-to-the-mean effect. In 1980, when George Brett batted .384, most people did not expect him to hit .384 the following year. When Wilt Chamberlain scored 100 points in a single game, most people did not expect him to score 100 points in his next game. When a historically 3.0 student got a 4.0 one semester, her friends did not expect a repeat performance the following semester. When a real estate agent sold five houses in one month (an abnormally high performance), his co-agents did not expect similar performance in the following month. Why is regression to the mean more intuitive in these cases? Because the performance is so extreme that we know it cannot last. Thus, under very unusual circumstances, we expect performance to regress. However, we generally do not recognize the regression effect in less extreme cases.

Consider Kahneman and Tversky's (1973) classic example in which the misconceptions surrounding regression led to overestimation of the effectiveness of punishment and the underestimation of the power of reward. Here, in a

discussion about flight training, experienced instructors noted that praise for an exceptionally smooth landing was typically followed by a poorer landing on the next try, while harsh criticism after a rough landing was usually followed by an improvement on the next try. The instructors concluded that verbal rewards were detrimental to learning, while verbal punishments were beneficial. Obviously, the tendency of performance to regress to the mean can account for the results; verbal feedback may have had absolutely no effect. However, to the extent that the instructors were prone to biased decision making, they were prone to reach the false conclusion that punishment is more effective than positive reinforcement in shaping behavior.

How do managers respond when they do not acknowledge the regression principle? Consider an employee with very high performance in one performance period. He (and his boss) may inappropriately expect similar performance in the next period. What happens when his performance regresses toward the mean? He (and his boss) begin to make excuses for not meeting expectations. Obviously, they are likely to develop false explanations and may inappropriately plan their future efforts.

Bias 8—The Conjunction Fallacy

Problem 9: Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and she participated in antinuclear demonstrations.

Rank order the following eight descriptions in terms of the probability (likelihood) that they describe Linda:

- ___ a. Linda is a teacher in an elementary school.
- ___ b. Linda works in a bookstore and takes yoga classes.
- ___ c. Linda is active in the feminist movement.
- ___ d. Linda is a psychiatric social worker.
- ___ e. Linda is a member of the League of Women Voters.
- ___ f. Linda is a bank teller.
- ___ g. Linda is an insurance salesperson.
- ___ h. Linda is a bank teller who is active in the feminist movement.

Examine your rank orderings of descriptions C, F, and H. Most people rank order C as more likely than H and H as more likely than F. The reason for this ordering is that C-H-F is the order of the degree to which the descriptions are *representative* of the short profile of Linda. The description of Linda was constructed by Tversky and Kahneman to be representative of an active feminist and unrepresentative of a bank teller. Recall from the representativeness heuristic that people make judgments according to the degree to which a specific description corresponds to a broader category within their minds.

Linda's description is more representative of a feminist than of a feminist bank teller, and is more representative of a feminist bank teller than of a bank teller. Thus, the representativeness heuristic accurately predicts that most individuals will rank order the items C-H-F.

Although the representativeness heuristic accurately predicts how individuals will respond, it also leads to another common, systematic distortion of human judgment—the **conjunction fallacy** (Tversky and Kahneman, 1983). This is illustrated by a reexamination of the potential descriptions of Linda. One of the simplest and most fundamental qualitative laws of probability is that a subset (for example, being a bank teller and a feminist) cannot be more likely than a larger set that completely includes the subset (e.g., being a bank teller). Statistically speaking, the broad set "Linda is a bank teller" must be rated at least as likely, if not more so, than the description "Linda is a bank teller and a feminist." After all, there is some chance (although it is small) that Linda is a bank teller but not a feminist. Based upon this logic, a rational assessment of the likelihoods of Linda being depicted by the eight descriptions must include a more likely rank for F than H.

While simple statistics can demonstrate that a conjunction (a combination of two or more descriptors) cannot be more probable than any one of its descriptors, the conjunction fallacy predicts and demonstrates that a conjunction will be judged more probable than a single component descriptor when the conjunction appears more representative than the component descriptor. Intuitively, thinking of Linda as a feminist bank teller "feels" more correct than thinking of her as only a bank teller.

The conjunction fallacy can also operate based on greater *availability* of the conjunction than one of the unique descriptors (Yates and Carlson, 1986). That is, if the conjunction creates more intuitive matches with vivid events, acts, or people than a component of the conjunction, the conjunction is likely to be perceived falsely as more probable than the component. For example, Tversky and Kahneman (1983) found experts (in July 1982) to evaluate the probability of

"a complete suspension of diplomatic relations between the USA and the Soviet Union, sometime in 1983"

as less likely than the probability of

"a Russian invasion of Poland, and a complete suspension of diplomatic relations between the USA and the Soviet Union, some time in 1983."

As earlier demonstrated, *suspension* is necessarily more likely than *invasion and suspension*. However, a Russian invasion followed by a diplomatic crisis provides a more intuitively viable story than simply a diplomatic crisis. Similarly, in the domain of natural disasters, Kahneman and Tversky's subjects rated

"a massive flood somewhere in North America in 1989, in which 1,000 people drown"

as less likely than the probability of

"an earthquake in California sometime in 1989, causing a flood in which more than 1,000 people drown."

It is obvious that the latter possibility is a subset of the former, and many other events could cause the flood in North America.

Tversky and Kahneman (1983) have shown that the conjunction fallacy is likely to lead to deviations from rationality in the judgments of sporting events, criminal behavior, international relations, and medical judgments. Our obvious concern with biased decision making resulting from the conjunction fallacy is that if we make systematic deviations from rationality in the prediction of future outcomes, we will be less prepared for dealing with future events.

Summary This discussion concludes our examination of the five biases (insensitivity to base rates, insensitivity to sample size, misconceptions of chance, regression to the mean, and the conjunction fallacy) that emanate from the use of the representativeness heuristic. Experience has taught us that the likelihood of a specific occurrence is related to the likelihood of a group of occurrences that that specific occurrence represents. Unfortunately, we tend to overuse this information in making decisions. The five biases we have just explored illustrate the systematic irrationalities that can occur in our judgments when we are not aware of this overreliance.

BIASES EMANATING FROM ANCHORING AND ADJUSTMENT

Bias 9—Insufficient Anchor Adjustment

Problem 10: A newly hired engineer for a computer firm in the Boston metropolitan area has four years of experience and good all-around qualifications. When asked to estimate the starting salary for this employee, my secretary (knowing very little about the profession or the industry) guessed an annual salary of \$23,000. What is your estimate?
\$ _____ per year.

Was your answer affected by my secretary's response? Most people do not think that my secretary's response affected their response. However, individuals are affected by the fairly irrelevant information contained in my secretary's estimate. Reconsider how you would have responded if my secretary had estimated \$80,000. On average, individuals give higher salary estimates to the problem when the secretary's estimate is stated as \$80,000 than when it is stated as \$23,000. Why? Studies have found that people develop estimates by starting from an initial anchor, based upon whatever information is provided, and adjusting from there to yield a final answer. Slovic and Lichtenstein (1971) have provided conclusive evidence that adjustments away from anchors are usually not sufficient to negate the effects of the anchor. In all cases, answers

are biased toward the initial anchor, even if it is irrelevant. Different starting points yield different answers. Tversky and Kahneman (1973) named this phenomenon **anchoring and adjustment**.

Tversky and Kahneman (1974) provide systematic, empirical evidence of the anchoring effect. For example, in one study, subjects were asked to estimate the percentage of African countries in the United Nations. For each subject, a *random* number (obtained by an observed spin of a roulette wheel) was given as a starting point. From there, subjects were asked to state whether the actual value of the quantity was higher or lower than this random value and then develop their best estimate for the actual quantity. It was found that the *arbitrary* values from the roulette wheel had a substantial impact on estimates. For example, for groups that received 10 countries and 65 countries as starting points, the median estimates were 25 and 45, respectively. Thus, even though the subjects were aware that the anchor was random and unrelated to the judgment task, the anchor had a dramatic effect on their judgment. Interestingly, paying subjects differentially based upon accuracy did not reduce the magnitude of the anchoring effect.

Salary negotiations represent a very common context for observing anchoring in the managerial world. For example, pay increases often come in the form of a percentage increase. A firm may have an average increase of 8 percent, with increases for specific employees varying from 3 percent to 13 percent. While society has led us to accept such systems as equitable, I believe that such a system falls victim to anchoring and leads to substantial inequities. What happens if an employee has been *substantially* underpaid to begin with? The pay system described does not rectify past inequities, since a pay increase of 11 percent will probably leave that employee still underpaid. Conversely, the system would work in the employee's favor had she been overpaid. It is common for an employer to ask job applicants their current salaries. Why? Employers are searching for a value from which they can anchor an adjustment. If the employee is worth far more than his current salary, the anchoring and adjustment hypothesis predicts that the firm will make an offer below the employee's true value. Does this figure provide fully accurate information about the true worth of the employee? I think not. Thus, the use of such compensation systems accepts past inequities as an anchor and makes inadequate adjustments from that point. Further, these findings suggest that in deciding what offer to make to a potential employee, any anchor that creeps into the discussion is likely to have an inappropriate effect on the eventual offer, even if the anchor is "ignored" as being ridiculous.

There are numerous examples of the anchoring-and-adjustment phenomenon in everyday life.

- In education, children are tracked by a school system that may categorize them into a certain level of performance at an early age. For example, a child who is anchored in the C group may meet expectations of mediocre performance. Conversely, a child of similar abilities anchored in the A track may strive to meet expectations, which will keep him in the A track.

- We have all fallen victim to the first-impression syndrome when meeting someone for the first time. We often place so much emphasis on first impressions that we do not adjust our opinion appropriately at a later date.
- Prior to 1973–1974, the speed limit on most interstate highways was 65 miles per hour (mph), with a normal cruising speed in the left-hand lane of 70 to 75 mph. This did not seem to be an extraordinarily unsafe speed to most people. After 1974, the speed limit was reduced to 55 mph. Most people changed their judgments to view a speed of 70 to 75 mph as extremely unsafe—“something only crazy kids would do.” Today, the reinstitution of the 65 mph limit on nonurban highways has rejustified the safety of the 70 to 75 mph speed.

In a fascinating study of anchoring and adjustment in the real estate market, Northcraft and Neale (1987) surveyed an association of real estate brokers, who indicated that they believed that they could assess the value of properties to within 5 percent of their true or appraised value. Further, they were unanimous in stating that they did not factor the listing price of the property into their personal estimate of its “true” value. Northcraft and Neale then asked four groups of professional real estate brokers and undergraduate students to estimate the value of a real house. Both brokers and students were randomly assigned to one of four experimental groups. In each group, all participants were given a 10-page packet of information about the house that was being sold. The packet included not only background on the house, but also considerable information about prices and characteristics of other houses in the area that had recently been sold. The only difference in the information given to the four groups was the listing price for the house, which was selected to be +11 percent, +4 percent, –4 percent, and –11 percent of the actual appraised value of the property. After reading the material, all participants toured the house, as well as the surrounding neighborhood. Participants were then asked for their estimate of the house’s price. The final results suggested that *both* brokers and students were *significantly* affected by the listing price (the anchor) in determining the value. While the students readily admitted the role that the listing price played in their decision-making process, the brokers flatly denied their use of the listing price as an anchor for their evaluations of the property—despite the evidence to the contrary. This study provides convincing data to indicate that even experts are susceptible to the anchoring bias. Furthermore, experts are less likely to realize their use of this bias in making decisions.

Joyce and Biddle (1981) have also provided empirical support for the anchoring-and-adjustment effect on practicing auditors of Big Eight accounting firms. Specifically, subjects in one condition were asked the following:

It is well known that many cases of management fraud go undetected even when competent annual audits are performed. The reason, of course, is that Generally Accepted Auditing Standards are not designed specifically to detect executive-level management fraud. We are interested in obtaining an estimate from practicing au-

ditors of the prevalence of executive-level management fraud as a first step in ascertaining the scope of the problem.

1. Based on your audit experience, is the incidence of significant executive-level management fraud more than 10 in each 1,000 firms (that is, 1 percent) audited by Big Eight accounting firms?
 - a. Yes, more than 10 in each 1,000 Big Eight clients have significant executive-level management fraud.
 - b. No, fewer than 10 in each 1,000 Big Eight clients have significant executive-level management fraud.
2. What is your estimate of the number of Big Eight clients per 1,000 that have significant executive-level management fraud?
(Fill in the blank below with the appropriate number.)
_____ in each 1,000 Big Eight clients have significant executive-level management fraud.

The second condition differed only in that subjects were asked whether the fraud incidence was more or less than 200 in each 1,000 audited, rather than 10 in 1,000. Subjects in the former condition estimated a fraud incidence of 16.52 per 1,000 on average, compared with an estimated fraud incidence of 43.11 per 1,000 in the second condition! Here, even professional auditors fell victim to anchoring and adjustment.

The tendency to make insufficient adjustments is a direct result of the anchoring-and-adjustment heuristic described in the first chapter. Interestingly, Nisbett and Ross (1980) present an argument that suggests that the anchoring-and-adjustment bias itself dictates that it will be very difficult to get you to change your decision-making strategies as a result of reading this book. They argue that each of the heuristics that we identify are currently serving as your cognitive anchors and are central to your current judgment processes. Thus, any cognitive strategy that I suggest must be presented and understood in a manner that will force you to break your existing cognitive anchors. Based on the evidence in this section, this should be a difficult challenge—but one that is important enough to be worth the effort!

Bias 10—Conjunctive and Disjunctive Events Bias

Problem 11: Which of the following appears most likely?
Which appears second most likely?

- a. Drawing a red marble from a bag containing 50 percent red marbles and 50 percent white marbles.
- b. Drawing a red marble seven times in succession, with replacement (a selected marble is put back in the bag before the next marble is se-

lected), from a bag containing 90 percent red marbles and 10 percent white marbles.

- c. Drawing at least one red marble in seven tries, with replacement, from a bag containing 10 percent red marbles and 90 percent white marbles.

The most common answer in ordering the preferences is B-A-C. Interestingly, the correct order of likelihood is C (52 percent), A (50 percent), B (48 percent)—the exact opposite of the most common intuitive pattern! This result illustrates a general bias to overestimate the probability of conjunctive events—events that must occur in conjunction with one another (Bar-Hillel, 1973)—and to underestimate the probability of disjunctive events—events that occur independently (Tversky and Kahneman, 1974). Thus, when multiple events all need to occur (problem B), we overestimate the true likelihood, while if only one of many events needs to occur (problem C), we underestimate the true likelihood.

Kahneman and Tversky (1974) explain these effects in terms of the anchoring-and-adjustment heuristic. They argue that the probability of any one event occurring (for example, drawing one red marble) provides a natural anchor for the judgment of the total probability. Since adjustment from an anchor is typically insufficient, the perceived likelihood of choice B stays inappropriately close to 90 percent, while the perceived probability of choice C stays inappropriately close to 10 percent.

How is each of these biases manifested in an applied context? The overestimation of conjunctive events is a powerful explanation of the timing problems in projects that require multistage planning. Individuals, businesses, and governments frequently fall victim to the conjunction-events bias in terms of timing and budgets. Public works projects seldom finish on time or on budget. New product ventures frequently take longer than expected.

Consider the following:

- You are planning a construction project that consists of five distinct components. Your schedule is tight, and every component must be on time in order to meet a contractual deadline. Will you meet this deadline?
- You are managing a consulting project that consists of six teams, each of which is analyzing a different alternative. The alternatives cannot be compared until all teams complete their portion. Will you meet the deadline?
- After three years of study, doctoral students typically dramatically overestimate the likelihood of completing their dissertations within a year. At this stage, they typically can tell you how long each remaining component will take. Why do they not finish in one year?

The underestimation of disjunctive events explains our surprise when an unlikely event occurs. As Tversky and Kahneman (1974) argue, "A complex system, such as a nuclear reactor or the human body, will malfunction if any of its essential component fails. Even when the likelihood of failure in each component is slight, the probability of an overall failure can be high if many compo-

nents are involved." In *Normal Accidents*, Perrow (1984) argues against the safety of technologies like nuclear reactors and DNA research. He fears that society significantly underestimates the likelihood of system failure because of our judgmental failure to realize the multitude of things that can go wrong in these incredibly complex and interactive systems.

The understanding of our underestimation of disjunctive events also has its positive side. Consider the following:

It's Monday evening (10:00 P.M.). You get a phone call telling you that you must be at the Chicago office by 9:30 A.M. the next morning. You call all five airlines that have flights that get into Chicago by 9:00 A.M. Each has one flight, and all the flights are booked. When you ask the probability of getting on each of the flights if you show up at the airport in the morning, you are disappointed to hear probabilities of 30 percent, 25 percent, 15 percent, 20 percent, and 25 percent. Consequently, you do not expect to get to Chicago in time.

In this case, the disjunctive bias leads you to expect the worst. In fact, if the probabilities given by the airlines are unbiased, and independent there is a 73 percent chance of getting on one of the flights (assuming that you can arrange to be at the right ticket counter at the right time)!

Bias 11—Overconfidence

Problem 12: Listed below are 10 uncertain quantities. Do not look up any information on these items. For each, write down your best estimate of the quantity. Next, put a lower and upper bound around your estimate, such that you are 98 percent confident that your range surrounds the actual quantity.

- _____ a. Mobil Oil's sales in 1987
- _____ b. IBM's assets in 1987
- _____ c. Chrysler's profit in 1987
- _____ d. The number of U.S. industrial firms in 1987 with sales greater than those of Consolidated Papers
- _____ e. The U.S. gross national product in 1945
- _____ f. The amount of taxes collected by the U.S. Internal Revenue Service in 1970
- _____ g. The length (in feet) of the Chesapeake Bay Bridge-Tunnel
- _____ h. The area (in square miles) of Brazil
- _____ i. The size of the black population of San Francisco in 1970
- _____ j. The dollar value of Canadian exports of lumber in 1977

How many of your 10 ranges will actually surround the true quantities? If you set your ranges so that you were 98 percent confident, you should expect to

correctly bound approximately 9.8 or 9 to 10 of the 10 quantities. Let's look at the correct answers: (a) \$51,223,000,000; (b) \$63,688,000,000; (c) \$1,289,700,000; (d) 381; (e) \$212,300,000,000; (f) \$195,722,096,497; (g) 93,203; (h) 3,286,470; (i) 96,078; (j) \$2,386,282,000.

How many of your ranges actually surrounded the true quantities? If you surround 9–10, we can conclude that you were appropriately confident in your estimation ability. Most people only surround between 3 (30 percent) and 7 (70 percent), despite claiming a 98 percent confidence that each of the ranges will surround the true value. Why? Most of us are *overconfident* in our estimation abilities and do not acknowledge the actual uncertainty that exists.

In Alpert and Raiffa's (1969) initial demonstration of overconfidence based upon 1,000 observations (100 subjects on 10 items), 42.6 percent of quantities fell outside 90% confidence ranges. Since then, overconfidence has been identified as a common judgmental pattern and demonstrated in a wide variety of settings. For example, Fischhoff, Slovic, and Lichtenstein (1977) found that subjects who assigned odds of 1,000:1 of being correct were correct only 81 to 88 percent of the time. For odds of 1,000,000:1, their answers were correct only 90 to 96 percent of the time! Hazard and Peterson (1973) identified overconfidence among members of the armed forces, while Cambridge and Shreckengost (1980) found extreme overconfidence in CIA agents.

The most well-established finding in the overconfidence literature is the tendency of people to be most overconfident of the correctness of their answers when asked to respond to questions of moderate to extreme difficulty (Fischhoff, Slovic, and Lichtenstein, 1977; Koriati, Lichtenstein, and Fischhoff, 1980; Lichtenstein and Fischhoff, 1977, 1980). That is, as subjects' knowledge of a question decreases, they do not correspondingly decrease their level of confidence (Nickerson and McGoldrick, 1965; Pitz, 1974). However, subjects typically demonstrate no overconfidence, and often some underconfidence, to questions with which they are familiar. Thus we should be most alert to overconfidence in areas outside of our expertise.

There is a large degree of controversy over the explanations of why overconfidence exists (see Lichtenstein, Fischhoff, and Phillips [1982] for an extensive discussion). Tversky and Kahneman (1974) explain overconfidence in terms of anchoring. Specifically, they argue that when individuals are asked to set a confidence range around an answer, their initial estimate serves as an anchor which biases their estimation of confidence intervals in both directions. As explained earlier, adjustments from an anchor are usually insufficient, resulting in an overly narrow confidence band.

In their review of the overconfidence literature, Lichtenstein, Fischhoff, and Phillips (1982) suggest two viable strategies for eliminating overconfidence. First, they have found that giving people feedback about their overconfidence based on their judgments has been moderately successful at reducing this bias. Second, Koriati, Lichtenstein, and Fischhoff (1980) found that asking people to explain why their answers might be wrong (or far off the mark) can decrease overconfidence by getting subjects to see contradictions in their judgment.

Why should you be concerned about overconfidence? After all, it has probably given you the courage in the past to attempt endeavors that have stretched your abilities. However, consider the following:

- You are a medical doctor and are considering performing a difficult operation. The patient's family needs to know the likelihood of his surviving the operation. You respond "95 percent." Are you guilty of malpractice if you tend to be overconfident in your projections of survival?
- You work for the Nuclear Regulatory Commission and are 99.9 percent confident that a reactor will not leak. Can we trust your confidence? If not, can we run the enormous risks of overconfidence in this domain?
- Your firm has been threatened with a multimillion dollar law suit. If you lose, your firm is out of business. You are 98 percent confident that the firm will not lose in court. Is this degree of certainty sufficient for you to recommend rejecting an out-of-court settlement? Based on what you know now, are you still comfortable with your 98 percent estimate?
- You have developed a market plan for a new product. You are so confident in your plan that you have not developed any contingencies for early market failure. The plan of attack falls apart. Will your overconfidence wipe out any hope of expediting changes in the marketing strategy?

In each of these examples, we have introduced serious problems that can result from the tendency to be overconfident. Thus, while confidence in your abilities is necessary for achievement in life, and perhaps to inspire confidence in others, you may want to monitor your overconfidence to achieve more effective professional decision making.

Summary The need for an initial anchor weighs strongly in our decision-making processes when we try to estimate likelihoods (such as the probability of on-time project completion) or establish values (like what salary to offer). Experience has taught us that starting from somewhere is easier than starting from nowhere in determining such figures. However, as the last three biases (insufficient anchor adjustment, conjunctive and disjunctive events bias, and overconfidence) show, we frequently overrely on these anchors and seldom question their validity or appropriateness in a particular situation. As with the other heuristics, we frequently fail even to realize that this heuristic is impacting our judgments.

TWO MORE GENERAL BIASES

Bias 12—The Confirmation Trap

Problem 13: (Adapted from Einhorn and Hogarth, 1978)

It is claimed that when a particular analyst predicts a rise in the market, the

market always rises. You are to check this claim. Examine the information available about the following four events (cards):

Card 1 Prediction: Favorable report	Card 2 Prediction: Unfavorable report	Card 3 Outcome: Rise in the market	Card 4 Outcome: Fall in the market
--	--	---	---

You currently see the predictions (cards 1 and 2) or outcomes (cards 3 and 4) associated with four events. You are seeing one side of a card. On the other side of cards 1 and 2 is the actual outcome, while on the other side of cards 3 and 4 is the prediction that the analyst made. Evidence about the claim is potentially available by turning over the card(s). Which cards would you turn over for the evidence that you need to check the analyst's claim? (Circle the appropriate cards.)

Consider the two most common responses: (1) "Card 1 (only)—that is the only card that I know has a favorable report and thus allows me to see whether a favorable report is actually followed by a rise in the market" and (2) "Cards 1 and 3—card 1 serves as a direct test, while card 3 allows me to see whether they made a favorable report when I know the market rose." Logical? Most people think that at least one of these two common responses is logical. However, both strategies demonstrate the tendency to search for confirming, rather than disconfirming, evidence. Einhorn and Hogarth (1978) argue that 1 and 4 is the correct answer to this quiz item. Why? Consider the following logic:

Card 1 allows me to test the claim that a rise in the market will add confirming evidence, while a fall in the market will fully disconfirm the claim, since the claim is that the market will *always* rise following a favorable report. Card 2 has no relevant information, since the claim does not address unfavorable reports by the analyst. While card 3 can add confirming evidence to card 1, it provides no unique information, since it cannot disconfirm the claim. That is, if an unfavorable report was made on card 3, then the event is not addressed by the claim. Finally, card 4 is critical. If it says "favorable report" on the other side, the claim is disconfirmed.

If you chose cards 1 and 3, you may have obtained a wealth of confirmatory information and were likely to inappropriately accept the claim. Only by including card 4 is there potential for disconfirmation of the hypothesis. Why do very few subjects select card 4? *Most of us seek confirmatory evidence and exclude the search for disconfirming information from our decision process.* However, it is typically not possible to know something to be true without checking for possible disconfirmation.

The initial demonstration of our tendency to ignore disconfirming information was provided in a series of projects by Wason (1960, 1968a, 1968b). In the first study, Wason (1960) presented subjects with the three-number sequence 2-4-6. The subject's task was to discover the numeric rule to which the three

numbers conformed. To determine the rule, subjects were allowed to generate other sets of three numbers that the experimenter would classify as either conforming or not conforming to the rule. At any point, subjects could stop when they thought that they had discovered the rule. How would you approach this problem?

Wason's rule was "any three ascending numbers"—a solution which required the accumulation of disconfirming, rather than confirming, evidence. For example, if you thought the rule included "the difference between the first two numbers equaling the difference between the last two numbers" (a common expectation), you must try sequences that do *not* conform to this rule to find the actual rule. Trying the sequences 1-2-3, 10-15-20, 122-126-130, and so on, will only lead you into the confirmation trap. In Wason's (1960) experiment, only 6 out of 29 subjects found the correct rule the first time that they thought they knew the answer. Wason concluded that obtaining the correct solution necessitates "a willingness to attempt to falsify hypotheses, and thus to test those intuitive ideas which so often carry the feeling of certitude" (p. 139).

This result was also observed by Einhorn and Hogarth (1978) with a sample of 23 statisticians. When that group responded to a problem very similar to problem 13, eleven asked for card 1; one asked for card 1 or 3; one asked for any one card; two asked for card 1 or 4; three asked for card 4 alone; and only five trained statisticians asked for cards 1 and 4. Thus, this group tended to realize the worthlessness of card 3 but failed to realize the importance of card 4. This leads to the conclusion that the tendency to exclude disconfirming information in the search process is not eliminated by the formal scientific training that is expected of statisticians.

It is easy to observe the confirmation trap in your decision-making processes. You make a tentative decision (to buy a new car, to hire a particular employee, to start research and development on a new product line). Do you search for data that support your decision before making the final commitment? Most of us do. However, the existence of the confirmation trap implies that the search for challenging, or disconfirming, evidence will provide the most useful insights. For example, in confirming your decision to hire a particular employee, it is probably easy to find supporting positive information on the individual, but in fact the key issue may be the degree to which negative information on this individual, as well as positive information on another potential applicant, also exists.

Bias 13—Hindsight

Consider the following scenarios:

- You are an avid football fan, and you are watching a critical game in which your team is behind 35–31. With three seconds left, and the ball on the opponent's three-yard line, the quarterback *unsuccessfully* calls a pass play into the corner of the endzone. You immediately respond, "I knew that he shouldn't have called that play."

- You are riding in an unfamiliar area, and your spouse is driving. You approach an unmarked fork in the road, and your spouse decides to go to the right. Four miles and fifteen minutes later, it is clear that you are lost. You blurt out, "I knew that you should have turned left at the fork."
- A manager who works for you hired a new supervisor last year. You were well aware of the choices he had at the time and allowed him to choose the new employee on his own. You have just received production data on every supervisor. The data on the new supervisor are terrible. You call in the manager and claim, "There was plenty of evidence that he (the supervisor) was not the man for the job."
- As director of marketing in a consumer-goods organization, you have just presented the results of an extensive six-month study on current consumer preferences for the products manufactured by your organization. After the conclusion of your presentation, a senior vice-president responds, "I don't know why we spent so much time and money to collect these data. I could have told you what the results were going to be."

Do you recognize yourself? Do you recognize someone else? Each scenario is representative of a phenomenon that has been named "the Monday morning quarterback syndrome" (Fischhoff, 1975b), "the knew-it-all-along effect" (Wood, 1978), "creeping determinism" (Fischhoff, 1975a, 1975b, 1980), and "the hindsight bias" (Fischhoff, 1975a, 1975b). This body of research demonstrates that people are typically not very good at recalling or reconstructing the way an uncertain situation appeared to them *before* finding out the results of the decision. What play would have you called? Did you *really* know that your spouse should have turned left? Was there *really* evidence that the selected supervisor was not the man for the job? Could the senior vice-president *really* have predicted the results of the survey? Perhaps our intuition is sometimes accurate, but we tend to overestimate what we knew and distort our beliefs about what we knew beforehand based upon what we later found out. The phenomenon occurs when people look back on the judgment of others, as well as of themselves.

Fischhoff has provided substantial evidence of the prevalence of the hindsight effect (1975a, 1975b, 1977; Fischhoff and Beyth, 1975; Slovic and Fischhoff, 1977). For example, Fischhoff (1975a) examined the differences between hindsight and foresight in the context of judging historical events and clinical instances. In one study, subjects were divided into five groups and asked to read a passage about the war between the British and Gurka forces in 1814. One group was not told the result of the war. The remaining four groups of subjects were told either that (1) the British won; (2) the Gurkas won; (3) a military stalemate was reached with no peace settlement; or (4) a military stalemate was reached with a peace settlement. Obviously, only one group was told the truthful outcome—(1) in this case. Each subject was then asked what his or her subjective assessments of the probability of each of the outcomes would have been without the benefit of knowing the reported outcome. Based upon this and other varied examples, the strong, consistent finding was

that knowledge of an outcome increases an individual's belief about the degree to which he or she would have predicted that outcome without the benefit of that knowledge.

A number of explanations of the hindsight effect have been offered. One of the most pervasive is to explain hindsight in terms of the heuristics discussed in this book (Tversky and Kahneman, 1974). Anchoring may contribute to this bias when individuals interpret their prior subjective judgments of probabilities of an event's occurring in reference to the anchor of knowing whether or not that outcome actually occurred. Since adjustments to anchors are known to be inadequate, hindsight knowledge can be expected to bias perceptions of what one thinks one knew in foresight. Further, to the extent that the various pieces of data on the event vary in terms of their support for the actual outcome, evidence that is consistent with the known outcome may become cognitively more salient and thus more *available* in memory (Slovic and Fischhoff, 1977). This will lead an individual to justify a claimed foresight in view of "the facts provided." Finally, the relevance of a particular piece of data may later be judged important to the extent to which it is *representative* of the final observed outcome.

Claiming that what has happened was predictable based on foresight knowledge puts us in a position of using hindsight to criticize another's foresight judgment. In the short run, hindsight has a number of advantages. In particular, it is very flattering to believe that your judgment is far better than it actually is! However, hindsight reduces our ability to learn from the past and to evaluate objectively the decisions of ourselves and others. Leading researchers in performance evaluation (cf. Feldman, 1981) and decision theory (cf. Einhorn and Hogarth, 1981) have argued that, where possible, individuals should be rewarded based on the process and logic of their decisions, not on the results. A decision maker who makes a high-quality decision that does not work out should be rewarded, not punished. The rationale for this argument is that the results are affected by a variety of factors outside the direct control of the decision maker. However, to the extent that we rely on results and the hindsight corresponding to them, we will inappropriately evaluate the logic used by the decision maker in terms of the outcomes that occurred, not the methods that were employed.

INTEGRATION AND COMMENTARY

Heuristics, or rules of thumb, are the cognitive tools we use to simplify decision making. The preceding pages have described 13 of the most common biases that result when we overrely on these judgmental heuristics. These biases are summarized in Table 2.2, along with their associated heuristics. Again, it should be emphasized that more than one heuristic can be operating on our decision-making processes at any one time. We have attempted to identify only the dominant heuristic affecting each bias. In the last two biases, their effects are so broad that it is difficult to even determine a dominant heuristic.

While the use of quiz items has emphasized the biases that our heuristics

Table 2.2 Summary of 13 Biases Presented in Chapter 2

Bias	Description
Biases Emanating from the Availability Heuristic	
1 Ease of recall	Individuals judge events that are more easily recalled from memory, based upon vividness or recency, to be more numerous than events of equal frequency whose instances are less easily recalled.
2 Retrievability	Individuals are biased in their assessments of the frequency of events based upon how their memory structures affect the search process.
3 Presumed associations	Individuals tend to overestimate the probability of two events co-occurring based upon the number of similar associations that are easily recalled, whether from experience or social influence.
Biases Emanating from the Representativeness Heuristic	
4 Insensitivity to base rates	Individuals tend to ignore base rates in assessing the likelihood of events when any other descriptive information is provided—even if it is irrelevant.
5 Insensitivity to sample size	Individuals frequently fail to appreciate the role of sample size in assessing the reliability of sample information.
6 Misconceptions of chance	Individuals expect that a sequence of data generated by a random process will look "random," even when the sequence is too short for those expectations to be statistically valid.
7 Regression to the mean	Individuals tend to ignore the fact that extreme events tend to regress to the mean on subsequent trials.
8 The conjunction fallacy	Individuals falsely judge that conjunctions (two events co-occurring) are more probable than a more global set of occurrences of which the conjunction is a subset.

(continued)

Table 2.2 (Continued)

Bias	Description
Biases Emanating from Anchoring and Adjustment	
9 Insufficient anchor adjustment	Individuals make estimates for values based upon an initial value (derived from past events, random assignment, or whatever information is available) and typically make insufficient adjustments from that anchor when establishing a final value.
10 Conjunctive and disjunctive events bias	Individuals exhibit a bias toward overestimating the probability of conjunctive events and underestimating the probability of disjunctive events.
11 Overconfidence	Individuals tend to be overconfident of the infallibility of their judgments when answering moderately to extremely difficult questions.
Two More General Biases	
12 The confirmation trap	Individuals tend to seek confirmatory information for what they think is true and neglect the search for disconfirmatory evidence.
13 Hindsight	After finding out whether or not an event occurred, individuals tend to overestimate the degree to which they would have predicted the correct outcome.

create, it should be stressed that, overall, the use of these heuristics results in far more adequate than inadequate decisions. Our minds adopt these heuristics because, on average, any loss in quality of decisions is outweighed by the time saved. However, we argue against blanket acceptance of heuristics based upon this logic. First, as we have demonstrated in this chapter, there are many instances in which the loss in the quality of decisions far outweighs the time saved by the use of the heuristics. Second, the foregoing logic suggests that we have voluntarily accepted tradeoffs associated with the use of heuristics. But in reality, we have not: Most of us are unaware of their existence and their on-going impact upon our decision making. The difficulty with heuristics is that we typically do not recognize that we are using them, and we consequently fail to distinguish between situations in which their use is more and less appropriate.

To emphasize the distinction between the legitimate and illegitimate uses of heuristics, reconsider problem 6. In that problem, subjects tend to predict that Mark is more likely to take a job in "management of the arts," despite the fact that the contextual data overwhelmingly favor "management consulting." The representativeness heuristic, in this case, prevents us from appropriately incorporating relevant base-rate data. However, if the choice of "management consulting" were replaced with another less common career path for an MBA from a prestigious university (such as management in the steel industry), then the representativeness heuristic is likely to lead to an accurate prediction. That is, when base-rate data are unavailable or irrelevant (that is, the choices have the same base-rate), the representativeness heuristic provides a reasonably good cognitive tool for matching Mark to his most likely career path. *The key to improved judgment, therefore, lies in learning to distinguish between appropriate and inappropriate uses of heuristics.* This chapter provides a start in learning to make this distinction.

This book's examination of biases and heuristics does not end here. In fact, in the next three chapters we will continue to examine biases and heuristics in the areas of risk, the escalation of commitment, and creativity. The latter part of the book will examine biases in the context of more complicated multiparty decision-making situations.

The first part of the paper discusses the importance of the role of the state in the development of the economy. It argues that the state should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The second part of the paper discusses the importance of the role of the private sector in the development of the economy. It argues that the private sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The third part of the paper discusses the importance of the role of the public sector in the development of the economy. It argues that the public sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The fourth part of the paper discusses the importance of the role of the private sector in the development of the economy. It argues that the private sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The fifth part of the paper discusses the importance of the role of the public sector in the development of the economy. It argues that the public sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The sixth part of the paper discusses the importance of the role of the private sector in the development of the economy. It argues that the private sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The seventh part of the paper discusses the importance of the role of the public sector in the development of the economy. It argues that the public sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The eighth part of the paper discusses the importance of the role of the private sector in the development of the economy. It argues that the private sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The ninth part of the paper discusses the importance of the role of the public sector in the development of the economy. It argues that the public sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health. The tenth part of the paper discusses the importance of the role of the private sector in the development of the economy. It argues that the private sector should play a leading role in the development of the economy, particularly in the areas of infrastructure, education, and health.

Genetic Algorithms in Search, Optimization, and Machine Learning

David E. Goldberg

The University of Alabama



ADDISON-WESLEY PUBLISHING COMPANY, INC.

Reading, Massachusetts • Menlo Park, California • Sydney
Don Mills, Ontario • Madrid • San Juan • New York • Singapore
Amsterdam • Wokingham, England • Tokyo • Bonn

The procedures and applications presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications.

Library of Congress Cataloging-in-Publication Data

Goldberg, David E. (David Edward), 1953—
Genetic algorithms in search, optimization, and machine learning.

Bibliography: p.
Includes index.

1. Combinatorial optimization. 2. Algorithms.
3. Machine learning. I. Title.
QA402.5.G635 1989 006.3'1 88-6276
ISBN 0-201-15767-5

Reprinted with corrections January, 1989

Copyright © 1989 by Addison-Wesley Publishing Company, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Published simultaneously in Canada.

12 13 14 15 16 17 18-MA-97 96 95 94

1 | A Gentle Introduction to Genetic Algorithms

In this chapter, we introduce genetic algorithms: what they are, where they came from, and how they compare to and differ from other search procedures. We illustrate how they work with a hand calculation, and we start to understand their power through the concept of a schema or similarity template.

WHAT ARE GENETIC ALGORITHMS?

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance.

Genetic algorithms have been developed by John Holland, his colleagues, and his students at the University of Michigan. The goals of their research have been twofold: (1) to abstract and rigorously explain the adaptive processes of natural systems, and (2) to design artificial systems software that retains the important mechanisms of natural systems. This approach has led to important discoveries in both natural and artificial systems science.

The central theme of research on genetic algorithms has been *robustness*, the balance between efficiency and efficacy necessary for survival in many differ-

ent environments. The implications of robustness for artificial systems are manifold. If artificial systems can be made more robust, costly redesigns can be reduced or eliminated. If higher levels of adaptation can be achieved, existing systems can perform their functions longer and better. Designers of artificial systems—both software and hardware, whether engineering systems, computer systems, or business systems—can only marvel at the robustness, the efficiency, and the flexibility of biological systems. Features for self-repair, self-guidance, and reproduction are the rule in biological systems, whereas they barely exist in the most sophisticated artificial systems.

Thus, we are drawn to an interesting conclusion: where robust performance is desired (and where is it not?), nature does it better; the secrets of adaptation and survival are best learned from the careful study of biological example. Yet we do not accept the genetic algorithm method by appeal to this beauty-of-nature argument alone. Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces. The primary monograph on the topic is Holland's (1975) *Adaptation in Natural and Artificial Systems*. Many papers and dissertations establish the validity of the technique in function optimization and control applications. Having been established as a valid approach to problems requiring efficient and effective search, genetic algorithms are now finding more widespread application in business, scientific, and engineering circles. The reasons behind the growing numbers of applications are clear. These algorithms are computationally simple yet powerful in their search for improvement. Furthermore, they are not fundamentally limited by restrictive assumptions about the search space (assumptions concerning continuity, existence of derivatives, unimodality, and other matters). We will investigate the reasons behind these attractive qualities; but before this, we need to explore the robustness of more widely accepted search procedures.

ROBUSTNESS OF TRADITIONAL OPTIMIZATION AND SEARCH METHODS

This book is not a comparative study of search and optimization techniques. Nonetheless, it is important to question whether conventional search methods meet our robustness requirements. The current literature identifies three main types of search methods: calculus-based, enumerative, and random. Let us examine each type to see what conclusions may be drawn without formal testing.

Calculus-based methods have been studied heavily. These subdivide into two main classes: indirect and direct. Indirect methods seek local extrema by solving the usually nonlinear set of equations resulting from setting the gradient of the objective function equal to zero. This is the multidimensional generalization of the elementary calculus notion of extremal points, as illustrated in Fig. 1.1. Given a smooth, unconstrained function, finding a possible peak starts by restricting search to those points with slopes of zero in all directions. On the other hand,

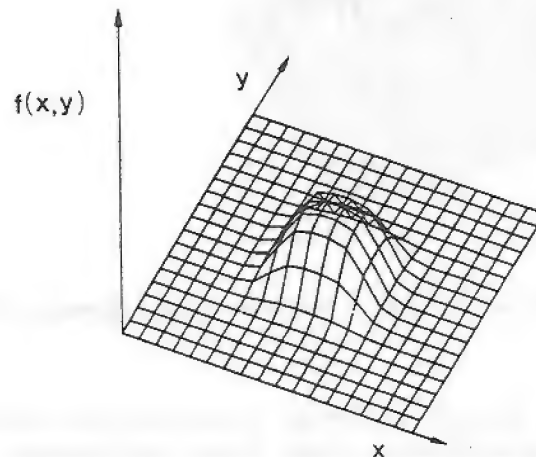


FIGURE 1.1 The single-peak function is easy for calculus-based methods.

direct (search) methods seek local optima by hopping on the function and moving in a direction related to the local gradient. This is simply the notion of *hill-climbing*: to find the local best, climb the function in the steepest permissible direction. While both of these calculus-based methods have been improved, extended, hashed, and rehashed, some simple reasoning shows their lack of robustness.

First, both methods are local in scope; the optima they seek are the best in a neighborhood of the current point. For example, suppose that Fig. 1.1 shows a portion of the complete domain of interest; a more complete picture is shown in Fig. 1.2. Clearly, starting the search or zero-finding procedures in the neighborhood of the lower peak will cause us to miss the main event (the higher peak). Furthermore, once the lower peak is reached, further improvement must be sought through random restart or other trickery. Second, calculus-based methods depend upon the existence of derivatives (well-defined slope values). Even if we allow numerical approximation of derivatives, this is a severe shortcoming. Many practical parameter spaces have little respect for the notion of a derivative and the smoothness this implies. Theorists interested in optimization have been too willing to accept the legacy of the great eighteenth and nineteenth-century mathematicians who painted a clean world of quadratic objective functions, ideal constraints, and ever present derivatives. The real world of search is fraught with discontinuities and vast multimodal, noisy search spaces as depicted in a less calculus-friendly function in Fig. 1.3. It comes as no surprise that methods depending upon the restrictive requirements of continuity and derivative existence are unsuitable for all but a very limited problem domain. For this reason and

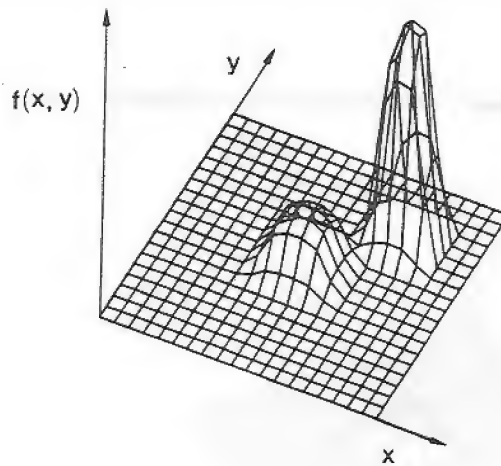


FIGURE 1.2 The multiple-peak function causes a dilemma. Which hill should we climb?

because of their inherently local scope of search, we must reject calculus-based methods. They are insufficiently robust in unintended domains.

Enumerative schemes have been considered in many shapes and sizes. The idea is fairly straightforward; within a finite search space, or a discretized infinite search space, the search algorithm starts looking at objective function values at every point in the space, one at a time. Although the simplicity of this type of

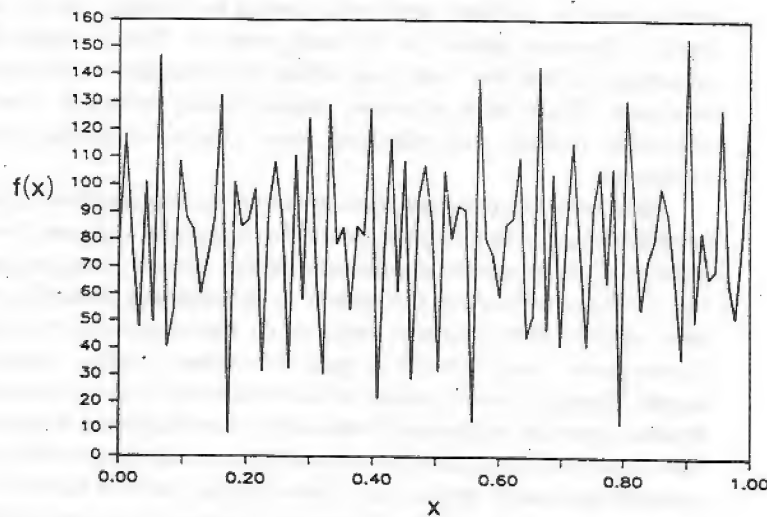


FIGURE 1.3 Many functions are noisy and discontinuous and thus unsuitable for search by traditional methods.

algorithm is attractive, and enumeration is a very human kind of search (when the number of possibilities is small), such schemes must ultimately be discounted in the robustness race for one simple reason: lack of efficiency. Many practical spaces are simply too large to search one at a time and still have a chance of using the information to some practical end. Even the highly touted enumerative scheme *dynamic programming* breaks down on problems of moderate size and complexity, suffering from a malady melodramatically labeled the "curse of dimensionality" by its creator (Bellman, 1961). We must conclude that less clever enumerative schemes are similarly, and more abundantly, cursed for real problems.

Random search algorithms have achieved increasing popularity as researchers have recognized the shortcomings of calculus-based and enumerative schemes. Yet, random walks and random schemes that search and save the best must also be discounted because of the efficiency requirement. Random searches, in the long run, can be expected to do no better than enumerative schemes. In our haste to discount strictly random search methods, we must be careful to separate them from randomized techniques. The genetic algorithm is an example of a search procedure that uses random choice as a tool to guide a highly exploitative search through a coding of a parameter space. Using random choice as a tool in a directed search process seems strange at first, but nature contains many examples. Another currently popular search technique, *simulated annealing*, uses random processes to help guide its form of search for minimal energy states. A recent book (Davis, 1987) explores the connections between simulated annealing and genetic algorithms. The important thing to recognize at this juncture is that randomized search does not necessarily imply directionless search.

While our discussion has been no exhaustive examination of the myriad methods of traditional optimization, we are left with a somewhat unsettling conclusion: conventional search methods are not robust. This does not imply that they are not useful. The schemes mentioned and countless hybrid combinations and permutations have been used successfully in many applications; however, as more complex problems are attacked, other methods will be necessary. To put this point in better perspective, inspect the problem spectrum of Fig. 1.4. In the figure a mythical effectiveness index is plotted across a problem continuum for a specialized scheme, an enumerative scheme, and an idealized robust scheme. The gradient technique performs well in its narrow problem class, as we expect, but it becomes highly inefficient (if useful at all) elsewhere. On the other hand, the enumerative scheme performs with egalitarian inefficiency across the spectrum of problems, as shown by the lower performance curve. Far more desirable would be a performance curve like the one labeled Robust Scheme. It would be worthwhile sacrificing peak performance on a particular problem to achieve a relatively high level of performance across the spectrum of problems. (Of course, with broad, efficient methods we can always create hybrid schemes that combine the best of the local search method with the more general robust scheme. We will have more to say about this possibility in Chapter 5.) We shall soon see how genetic algorithms help fill this robustness gap.

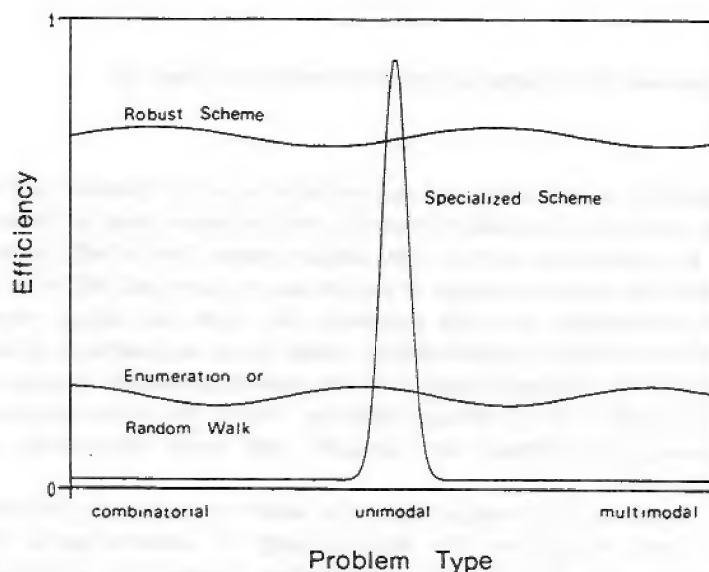


FIGURE 1.4 Many traditional schemes work well in a narrow problem domain. Enumerative schemes and random walks work equally inefficiently across a broad spectrum. A robust method works well across a broad spectrum of problems.

THE GOALS OF OPTIMIZATION

Before examining the mechanics and power of a simple genetic algorithm, we must be clearer about our goals when we say we want to optimize a function or a process. What are we trying to accomplish when we optimize? The conventional view is presented well by Beightler, Phillips, and Wilde (1979, p. 1):

Man's longing for perfection finds expression in the theory of optimization. It studies how to describe and attain what is Best, once one knows how to measure and alter what is Good or Bad. . . . *Optimization theory* encompasses the quantitative study of optima and methods for finding them.

Thus optimization seeks to improve performance toward some optimal point or points. Note that this definition has two parts: (1) we seek improvement to approach some (2) optimal point. There is a clear distinction between the *process* of improvement and the *destination* or optimum itself. Yet, in judging optimization procedures we commonly focus solely upon convergence (does the method reach the optimum?) and forget entirely about interim performance. This emphasis stems from the origins of optimization in the calculus. It is not, however, a natural emphasis.

Consider a human decision maker, for example, a businessman. How do we judge his decisions? What criteria do we use to decide whether he has done a good or bad job? Usually we say he has done well when he makes adequate selections within the time and resources allotted. Goodness is judged relative to his competition. Does he produce a better widget? Does he get it to market more efficiently? With better promotion? We never judge a businessman by an attainment-of-the-best criterion; perfection is all too stern a taskmaster. As a result, we conclude that convergence to the best is not an issue in business or in most walks of life; we are only concerned with doing better relative to others. Thus, if we want more humanlike optimization tools, we are led to a reordering of the priorities of optimization. The most important goal of optimization is improvement. Can we get to some good, "satisficing" (Simon, 1969) level of performance quickly? Attainment of the optimum is much less important for complex systems. It would be nice to be perfect; meanwhile, we can only strive to improve. In the next chapter we watch the genetic algorithm for these qualities; here we outline some important differences between genetic algorithms and more traditional methods.

HOW ARE GENETIC ALGORITHMS DIFFERENT FROM TRADITIONAL METHODS?

In order for genetic algorithms to surpass their more traditional cousins in the quest for robustness, GAs must differ in some very fundamental ways. Genetic algorithms are different from more normal optimization and search procedures in four ways:

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

Genetic algorithms require the natural parameter set of the optimization problem to be coded as a finite-length string over some finite alphabet. As an example, consider the optimization problem posed in Fig. 1.5. We wish to maximize the function $f(x) = x^2$ on the integer interval $[0, 31]$. With more traditional methods we would be tempted to twiddle with the parameter x , turning it like the vertical hold knob on a television set, until we reached the highest objective function value. With GAs, the first step of our optimization process is to code the parameter x as a finite-length string. There are many ways to code the x parameter, and Chapter 3 examines some of these in detail. At the moment, let's consider an optimization problem where the coding comes a bit more naturally.

Consider the black box switching problem illustrated in Fig. 1.6. This problem concerns a black box device with a bank of five input switches. For every setting of the five switches, there is an output signal f , mathematically $f = f(s)$,

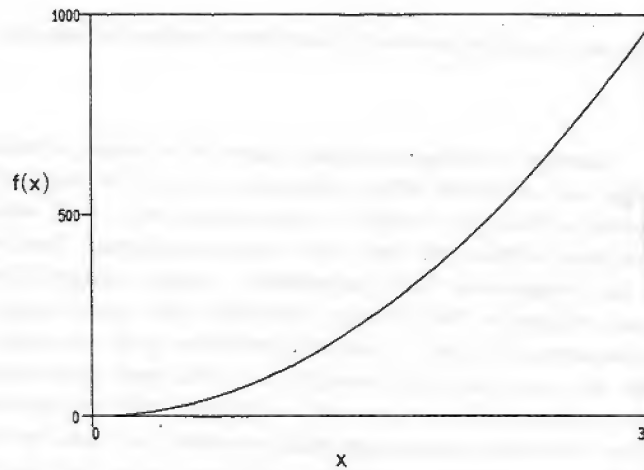


FIGURE 1.5 A simple function optimization example, the function $f(x) = x^2$ on the integer interval $[0, 31]$.

where s is a particular setting of the five switches. The objective of the problem is to set the switches to obtain the maximum possible f value. With other methods of optimization we might work directly with the parameter set (the switch settings) and toggle switches from one setting to another using the transition rules of our particular method. With genetic algorithms, we first code the switches as a finite-length string. A simple code can be generated by considering a string of five 1's and 0's where each of the five switches is represented by a 1 if the switch is on and a 0 if the switch is off. With this coding, the string 11110 codes the setting where the first four switches are on and the fifth switch is off. Some of the codings introduced later will not be so obvious, but at this juncture we acknowledge that genetic algorithms use codings. Later it will be apparent

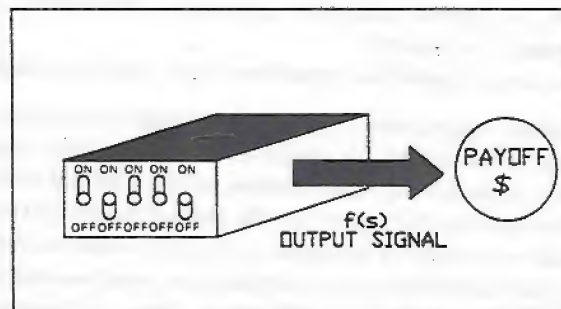


FIGURE 1.6 A black box optimization problem with five on-off switches illustrates the idea of a coding and a payoff measure. Genetic algorithms only require these two things: they don't need to know the workings of the black box.

that genetic algorithms exploit coding similarities in a very general way; as a result, they are largely unconstrained by the limitations of other methods (continuity, derivative existence, unimodality, and so on).

In many optimization methods, we move gingerly from a single point in the decision space to the next using some transition rule to determine the next point. This point-to-point method is dangerous because it is a perfect prescription for locating false peaks in multimodal (many-peaked) search spaces. By contrast, GAs work from a rich database of points simultaneously (a population of strings), climbing many peaks in parallel; thus, the probability of finding a false peak is reduced over methods that go point to point. As an example, let's consider our black box optimization problem (Fig. 1.6) again. Other techniques for solving this problem might start with one set of switch settings, apply some transition rules, and generate a new trial switch setting. A genetic algorithm starts with a population of strings and thereafter generates successive populations of strings. For example, in the five-switch problem, a random start using successive coin flips (head = 1, tail = 0) might generate the initial population of size $n = 4$ (small by genetic algorithm standards):

```
01101
11000
01000
10011
```

After this start, successive populations are generated using the genetic algorithm. By working from a population of well-adapted diversity instead of a single point, the genetic algorithm adheres to the old adage that there is safety in numbers: we will soon see how this parallel flavor contributes to a genetic algorithm's robustness.

Many search techniques require much auxiliary information in order to work properly. For example, gradient techniques need derivatives (calculated analytically or numerically) in order to be able to climb the current peak, and other local search procedures like the greedy techniques of combinatorial optimization (Lawler, 1976; Syslo, Deo, and Kowalik, 1983) require access to most if not all tabular parameters. By contrast, genetic algorithms have no need for all this auxiliary information: GAs are blind. To perform an effective search for better and better structures, they only require payoff values (objective function values) associated with individual strings. This characteristic makes a GA a more canonical method than many search schemes. After all, every search problem has a metric (or metrics) relevant to the search; however, different search problems have vastly different forms of auxiliary information. Only if we refuse to use this auxiliary information can we hope to develop the broadly based schemes we desire. On the other hand, the refusal to use specific knowledge when it does exist can place an upper bound on the performance of an algorithm when it goes head to head with methods designed for that problem. Chapter 5 examines ways to use nonpayoff information in so-called knowledge-directed genetic algorithms; however, at this juncture we stress the importance of the blindness assumption to pure genetic algorithm robustness.

Unlike many methods, GAs use probabilistic transition rules to guide their search. To persons familiar with deterministic methods this seems odd, but the use of probability does not suggest that the method is some simple random search; this is not decision making at the toss of a coin. Genetic algorithms use random choice as a tool to guide a search toward regions of the search space with likely improvement.

Taken together, these four differences—direct use of a coding, search from a population, blindness to auxiliary information, and randomized operators—contribute to a genetic algorithm's robustness and resulting advantage over other more commonly used techniques. The next section introduces a simple three-operator genetic algorithm.

A SIMPLE GENETIC ALGORITHM

The mechanics of a simple genetic algorithm are surprisingly simple, involving nothing more complex than copying strings and swapping partial strings. The explanation of why this simple process works is much more subtle and powerful. Simplicity of operation and power of effect are two of the main attractions of the genetic algorithm approach.

The previous section pointed out how genetic algorithms process populations of strings. Recalling the black box switching problem, remember that the initial population had four strings:

```
01101
11000
01000
10011
```

Also recall that this population was chosen at random through 20 successive flips of an unbiased coin. We now must define a set of simple operations that take this initial population and generate successive populations that (we hope) improve over time.

A simple genetic algorithm that yields good results in many practical problems is composed of three operators:

1. Reproduction
2. Crossover
3. Mutation

Reproduction is a process in which individual strings are copied according to their objective function values, f (biologists call this function the fitness function). Intuitively, we can think of the function f as some measure of profit, utility, or goodness that we want to maximize. Copying strings according to their fitness values means that strings with a higher value have a higher probability of contributing one or more offspring in the next generation. This operator, of course, is an artificial version of natural selection, a Darwinian survival of the fittest

TABLE 1.1 Sample Problem Strings and Fitness Values

No.	String	Fitness	% of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

among string creatures. In natural populations fitness is determined by a creature's ability to survive predators, pestilence, and the other obstacles to adulthood and subsequent reproduction. In our unabashedly artificial setting, the objective function is the final arbiter of the string-creature's life or death.

The reproduction operator may be implemented in algorithmic form in a number of ways. Perhaps the easiest is to create a biased roulette wheel where each current string in the population has a roulette wheel slot sized in proportion to its fitness. Suppose the sample population of four strings in the black box problem has objective or fitness function values f as shown in Table 1.1 (for now we accept these values as the output of some unknown and arbitrary black box—later we will examine a function and coding that generate these same values).

Summing the fitness over all four strings, we obtain a total of 1170. The percentage of population total fitness is also shown in the table. The corresponding weighted roulette wheel for this generation's reproduction is shown in Fig. 1.7. To reproduce, we simply spin the weighted roulette wheel thus defined four times. For the example problem, string number 1 has a fitness value of 169, which represents 14.4 percent of the total fitness. As a result, string 1 is given 14.4 percent of the biased roulette wheel, and each spin turns up string 1 with prob-

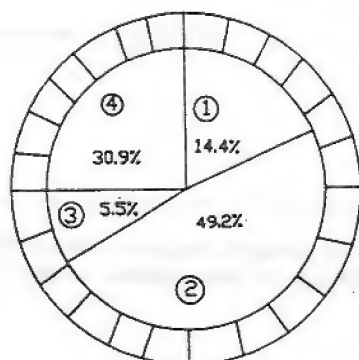


FIGURE 1.7 Simple reproduction allocates offspring strings using a roulette wheel with slots sized according to fitness. The sample wheel is sized for the problem of Tables 1.1 and 1.2.

ability 0.144. Each time we require another offspring, a simple spin of the weighted roulette wheel yields the reproduction candidate. In this way, more highly fit strings have a higher number of offspring in the succeeding generation. Once a string has been selected for reproduction, an exact replica of the string is made. This string is then entered into a mating pool, a tentative new population, for further genetic operator action.

After reproduction, simple crossover (Fig. 1.8) may proceed in two steps. First, members of the newly reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over as follows: an integer position k along the string is selected uniformly at random between 1 and the string length less one $[1, l - 1]$. Two new strings are created by swapping all characters between positions $k + 1$ and l inclusively. For example, consider strings A_1 and A_2 from our example initial population:

$$\begin{aligned} A_1 &= 0 \ 1 \ 1 \ 0 \ | \ 1 \\ A_2 &= 1 \ 1 \ 0 \ 0 \ | \ 0 \end{aligned}$$

Suppose in choosing a random number between 1 and 4, we obtain a $k = 4$ (as indicated by the separator symbol $|$). The resulting crossover yields two new strings where the prime (') means the strings are part of the new generation:

$$\begin{aligned} A'_1 &= 0 \ 1 \ 1 \ 0 \ 0 \\ A'_2 &= 1 \ 1 \ 0 \ 0 \ 1 \end{aligned}$$

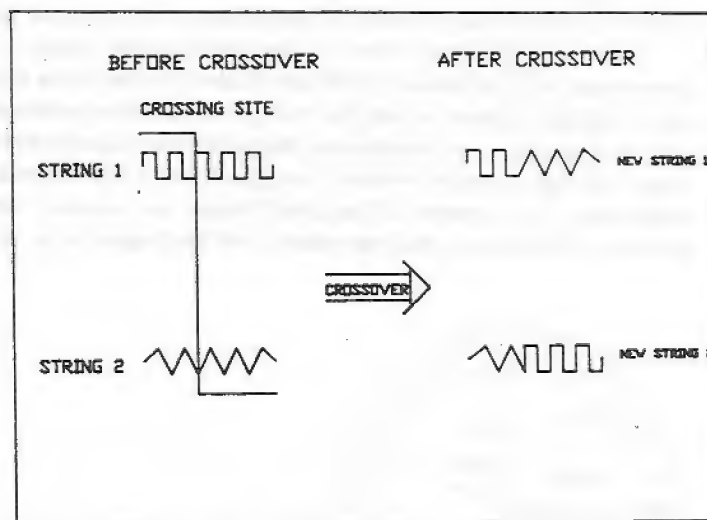


FIGURE 1.8 A schematic of simple crossover shows the alignment of two strings and the partial exchange of information, using a cross site chosen at random.

The mechanics of reproduction and crossover are surprisingly simple, involving random number generation, string copies, and some partial string exchanges. Nonetheless, the combined emphasis of reproduction and the structured, though randomized, information exchange of crossover give genetic algorithms much of their power. At first this seems surprising. How can two such simple (and computationally trivial) operators result in anything useful, let alone a rapid and robust search mechanism? Furthermore, doesn't it seem a little strange that chance should play such a fundamental role in a directed search process? We will examine a partial answer to the first of these two questions in a moment; the answer to the second question was well recognized by the mathematician J. Hadamard (1949, p. 29):

We shall see a little later that the possibility of imputing discovery to pure chance is already excluded. . . . On the contrary, that there is an intervention of chance but also a necessary work of unconsciousness, the latter implying and not contradicting the former. . . . Indeed, it is obvious that invention or discovery, be it in mathematics or anywhere else, takes place by combining ideas.

Hadamard suggests that even though discovery is not a result—cannot be a result—of pure chance, it is almost certainly guided by directed serendipity. Furthermore, Hadamard hints that a proper role for chance in a more humanlike discovery mechanism is to cause the juxtaposition of different notions. It is interesting that genetic algorithms adopt Hadamard's mix of direction and chance in a manner that efficiently builds new solutions from the best partial solutions of previous trials.

To see this, consider a population of n strings (perhaps the four-string population for the black box problem) over some appropriate alphabet, coded so that each is a complete *idea* or prescription for performing a particular task (in this case, each string is one complete switch-setting idea). Substrings within each string (idea) contain various *notions* of what is important or relevant to the task. Viewed in this way, the population contains not just a sample of n ideas; rather, it contains a multitude of notions and rankings of those notions for task performance. Genetic algorithms ruthlessly exploit this wealth of information by (1) reproducing high-quality notions according to their performance and (2) crossing these notions with many other high-performance notions from other strings. Thus, the action of crossover with previous reproduction speculates on new ideas constructed from the high-performance building blocks (notions) of past trials. In passing, we note that despite the somewhat fuzzy definition of a notion, we have not limited a notion to simple linear combinations of single features or pairs of features. Biologists have long recognized that evolution must efficiently process the epistasis (positionwise nonlinearity) that arises in nature. In a similar manner, the notion processing of genetic algorithms must effectively process notions even when they depend upon their component features in highly nonlinear and complex ways.

Exchanging of notions to form new ideas is appealing intuitively, if we think in terms of the process of *innovation*. What is an innovative idea? As Hadamard suggests, most often it is a juxtaposition of things that have worked well in the past. In much the same way, reproduction and crossover combine to search potentially pregnant new ideas. This experience of emphasis and crossing is analogous to the human interaction many of us have observed at a trade show or scientific conference. At a widget conference, for example, various widget experts from around the world gather to discuss the latest in widget technology. After the lecture sessions, they all pair off around the bar to exchange widget stories. Well-known widget experts, of course, are in greater demand and exchange more ideas, thoughts, and notions with their lesser known widget colleagues. When the show ends, the widget people return to their widget laboratories to try out a surfeit of widget innovations. The process of reproduction and crossover in a genetic algorithm is this kind of exchange. High-performance notions are repeatedly tested and exchanged in the search for better and better performance.

If reproduction according to fitness combined with crossover gives genetic algorithms the bulk of their processing power, what then is the purpose of the mutation operator? Not surprisingly, there is much confusion about the role of mutation in genetics (both natural and artificial). Perhaps it is the result of too many B movies detailing the exploits of mutant eggplants that consume mass quantities of Tokyo or Chicago, but whatever the cause for the confusion, we find that mutation plays a decidedly secondary role in the operation of genetic algorithms. Mutation is needed because, even though reproduction and crossover effectively search and recombine extant notions, occasionally they may become overzealous and lose some potentially useful genetic material (1's or 0's at particular locations). In artificial genetic systems, the mutation operator protects against such an irrecoverable loss. In the simple GA, mutation is the occasional (with small probability) random alteration of the value of a string position. In the binary coding of the black box problem, this simply means changing a 1 to a 0 and vice versa. By itself, mutation is a random walk through the string space. When used sparingly with reproduction and crossover, it is an insurance policy against premature loss of important notions.

That the mutation operator plays a secondary role in the simple GA, we simply note that the frequency of mutation to obtain good results in empirical genetic algorithm studies is on the order of one mutation per thousand bit (position) transfers. Mutation rates are similarly small (or smaller) in natural populations, leading us to conclude that mutation is appropriately considered as a secondary mechanism of genetic algorithm adaptation.

Other genetic operators and reproductive plans have been abstracted from the study of biological example. However, the three examined in this section, reproduction, simple crossover, and mutation, have proved to be both computationally simple and effective in attacking a number of important optimization problems. In the next section, we perform a hand simulation of the simple genetic algorithm to demonstrate both its mechanics and its power.

GENETIC ALGORITHMS AT WORK—A SIMULATION BY HAND

Let's apply our simple genetic algorithm to a particular optimization problem step by step. Consider the problem of maximizing the function $f(x) = x^2$, where x is permitted to vary between 0 and 31, a function displayed earlier as Fig. 1.5. To use a genetic algorithm we must first code the decision variables of our problem as some finite-length string. For this problem, we will code the variable x simply as a binary unsigned integer of length 5. Before we proceed with the simulation, let's briefly review the notion of a binary integer. As decadigitated creatures, we have little problem handling base 10 integers and arithmetic. For example, the five-digit number 53,095 may be thought of as

$$5 \cdot 10^4 + 3 \cdot 10^3 + 0 \cdot 10^2 + 9 \cdot 10^1 + 5 \cdot 1 = 53,095.$$

In base 2 arithmetic, we of course only have two digits to work with, 0 and 1, and as an example the number 10,011 decodes to the base 10 number

$$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 2 + 1 = 19.$$

With a five-bit (*binary digit*) unsigned integer we can obtain numbers between 0 (00000) and 31 (11111). With a well-defined objective function and coding, we now simulate a single generation of a genetic algorithm with reproduction, crossover, and mutation.

To start off, we select an initial population at random. We select a population of size 4 by tossing a fair coin 20 times. We can skip this step by using the initial population created in this way earlier for the black box switching problem. Looking at this population, shown on the left-hand side of Table 1.2, we observe that the decoded x values are presented along with the fitness or objective function values $f(x)$. To make sure we know how the fitness values $f(x)$ are calculated from the string representation, let's take a look at the third string of the initial population, string 01000. Decoding this string as an unsigned binary integer, we note that there is a single one in the $2^3 = 8$'s position. Hence for string 01000 we obtain $x = 8$. To calculate the fitness or objective function we simply square the x value and obtain the resulting fitness value $f(x) = 64$. Other x and $f(x)$ values may be obtained similarly.

You may notice that the fitness or objective function values are the same as the black box values (compare Tables 1.1 and 1.2). This is no coincidence, and the black box optimization problem was well represented by the particular function, $f(x)$, and coding we are now using. Of course, the genetic algorithm need not know any of this; it is just as happy to optimize some arbitrary switching function (or any other finite coding and function for that matter) as some polynomial function with straightforward binary coding. This discussion simply reinforces one of the strengths of the genetic algorithm: by exploiting similarities in codings, genetic algorithms can deal effectively with a broader class of functions than can many other procedures.

A generation of the genetic algorithm begins with reproduction. We select the mating pool of the next generation by spinning the weighted roulette wheel

TABLE 1.2 A Genetic Algorithm by Hand

String No.	Initial Population (Randomly Generated)	x Value (Unsigned Integer)	$f(x)$ x^2	$p_{\text{select}},$ $\frac{f_i}{\Sigma f}$	Expected count $\frac{f_i}{\bar{f}}$	Actual Count from (Roulette Wheel)
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4.0
Average			<u>293</u>	0.25	1.00	1.0
Max			<u>576</u>	0.49	1.97	2.0

(shown in Fig. 1.7) four times. Actual simulation of this process using coin tosses has resulted in string 1 and string 4 receiving one copy in the mating pool, string 2 receiving two copies, and string 3 receiving no copies, as shown in the center of Table 1.2. Comparing this with the expected number of copies ($n \cdot p_{\text{select}}$) we have obtained what we should expect: the best get more copies, the average stay even, and the worst die off.

With an active pool of strings looking for mates, simple crossover proceeds in two steps: (1) strings are mated randomly, using coin tosses to pair off the happy couples, and (2) mated string couples cross over, using coin tosses to select the crossing sites. Referring again to Table 1.2, random choice of mates has selected the second string in the mating pool to be mated with the first. With a crossing site of 4, the two strings 01101 and 11000 cross and yield two new strings 01100 and 11001. The remaining two strings in the mating pool are crossed at site 2; the resulting strings may be checked in the table.

The last operator, mutation, is performed on a bit-by-bit basis. We assume that the probability of mutation in this test is 0.001. With 20 transferred bit positions we should expect $20 \cdot 0.001 = 0.02$ bits to undergo mutation during a given generation. Simulation of this process indicates that no bits undergo mutation for this probability value. As a result, no bit positions are changed from 0 to 1 or vice versa during this generation.

Following reproduction, crossover, and mutation, the new population is ready to be tested. To do this, we simply decode the new strings created by the simple genetic algorithm and calculate the fitness function values from the x values thus decoded. The results of a single generation of the simulation are shown at the right of Table 1.2. While drawing concrete conclusions from a single trial of a stochastic process is, at best, a risky business, we start to see how genetic algorithms combine high-performance notions to achieve better performance. In the table, note how both the maximal and average performance have improved in the new population. The population average fitness has improved from 293 to

TABLE 1.2 (Continued)

Mating Pool after Reproduction (Cross Site Shown)	Mate (Randomly Selected)	Crossover Site (Randomly Selected)	New Population	x Value	$f(x)$ x^2
0 1 1 0 1	2	4	0 1 1 0 0	12	144
1 1 0 0 0	1	4	1 1 0 0 1	25	625
1 1 0 0 0	4	2	1 1 0 1 1	27	729
1 0 0 1 1	3	2	1 0 0 0 0	16	256
					1754
					439
					<u>729</u>

NOTES:

- 1) Initial population chosen by four repetitions of five coin tosses where heads = 1, tails = 0.
- 2) Reproduction performed through 1 part in 8 simulation of roulette wheel selection (three coin tosses).
- 3) Crossover performed through binary decoding of 2 coin tosses (TT = 00, = 0 = cross site 1, HH = 11, = 3 = cross site 4).
- 4) Crossover probability assumed to be unity $p_c = 1.0$.
- 5) Mutation probability assumed to be 0.001, $p_m = 0.001$. Expected mutations = $5 \cdot 4 \cdot 0.001 = 0.02$. No mutations expected during a single generation. None simulated.

439 in one generation. The maximum fitness has increased from 576 to 729 during that same period. Although random processes help cause these happy circumstances, we start to see that this improvement is no fluke. The best string of the first generation (11000) receives two copies because of its high, above-average performance. When this combines at random with the next highest string (10011) and is crossed at location 2 (again at random), one of the resulting strings (11011) proves to be a very good choice indeed.

This event is an excellent illustration of the ideas and notions analogy developed in the previous section. In this case, the resulting good idea is the combination of two above-average notions, namely the substrings 11--- and ---11. Although the argument is still somewhat heuristic, we start to see how genetic algorithms effect a robust search. In the next section, we expand our understanding of these concepts by analyzing genetic algorithms in terms of schemata or similarity templates.

The intuitive viewpoint developed thus far has much appeal. We have compared the genetic algorithm with certain human search processes commonly called innovative or creative. Furthermore, hand simulation of the simple genetic algorithm has given us some confidence that indeed something interesting is going on here. Yet, something is missing. What is being processed by genetic algorithms and how do we know whether processing it (whatever it is) will lead to optimal or near optimal results in a particular problem? Clearly, as scientists,

engineers, and business managers we need to understand the what and the how of genetic algorithm performance.

To obtain this understanding, we examine the raw data available for any search procedure and discover that we can search more effectively if we exploit important similarities in the coding we use. This leads us to develop the important notion of a *similarity template*, or *schema*. This in turn leads us to a keystone of the genetic algorithm approach, the *building block hypothesis*.

GRIST FOR THE SEARCH MILL—IMPORTANT SIMILARITIES

For much too long we have ignored a fundamental question. In a search process given only payoff data (fitness values), what information is contained in a population of strings and their objective function values to help guide a directed search for improvement? To ask this question more clearly, consider the strings and fitness values originally displayed in Table 1.1 from the simulation of the previous section (the black box problem) and gathered below for convenience:

String	Fitness
01101	169
11000	576
01000	64
10011	361

What information is contained in this population to guide a directed search for improvement? On the face of it, there is not very much: four independent samples of different strings with their fitness values. As we stare at the page, however, quite naturally we start scanning up and down the string column, and we notice certain similarities among the strings. Exploring these similarities in more depth, we notice that certain string patterns seem highly associated with good performance. The longer we stare at the strings and their fitness values, the greater is the temptation to experiment with these high fitness associations. It seems perfectly reasonable to play mix and match with some of the substrings that are highly correlated with past success. For example, in the sample population, the strings starting with a 1 seem to be among the best. Might this be an important ingredient in optimizing this function? Certainly with our function ($f(x) = x^2$) and our coding (a five-bit unsigned integer) we know it is (why is this true?). But, what are we doing here? Really, two separate things. First, we are seeking similarities among strings in the population. Second, we are looking for causal relationships between these similarities and high fitness. In so doing, we admit a wealth of new information to help guide a search. To see how much and precisely

what information we admit, let us consider the important concept of a schema (plural, *schemata*), or similarity template.

SIMILARITY TEMPLATES (SCHEMATA)

In some sense we are no longer interested in strings as strings alone. Since important similarities among highly fit strings can help guide a search, we question how one string can be similar to its fellow strings. Specifically we ask, in what ways is a string a representative of other string classes with similarities at certain string positions? The framework of schemata provides the tool to answer these questions.

A schema (Holland, 1968, 1975) is a similarity template describing a subset of strings with similarities at certain string positions. For this discussion, let us once again limit ourselves without loss of generality to the binary alphabet $\{0,1\}$. We motivate a schema most easily by appending a special symbol to this alphabet; we add the $*$ or *don't care* symbol. With this extended alphabet we can now create strings (schemata) over the ternary alphabet $\{0, 1, *\}$, and the meaning of the schema is clear if we think of it as a pattern matching device: a schema matches a particular string if at every location in the schema a 1 matches a 1 in the string, a 0 matches a 0, or a $*$ matches either. As an example, consider the strings and schemata of length 5. The schema $*0000$ matches two strings, namely $\{10000, 00000\}$. As another example, the schema $*111*$ describes a subset with four members $\{01110, 01111, 11110, 11111\}$. As one last example, the schema $0*1**$ matches any of the eight strings of length 5 that begin with a 0 and have a 1 in the third position. As you can start to see, the idea of a schema gives us a powerful and compact way to talk about all the well-defined similarities among finite-length strings over a finite alphabet. We should emphasize that the $*$ is only a metasympol (a symbol about other symbols); it is never explicitly processed by the genetic algorithm. It is simply a notational device that allows description of all possible similarities among strings of a particular length and alphabet.

Counting the total number of possible schemata is an enlightening exercise. In the previous example, with $l = 5$, we note there are $3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 3^5 = 243$ different similarity templates because each of the five positions may be a 0, 1, or $*$. In general, for alphabets of cardinality (number of alphabet characters) k , there are $(k + 1)^l$ schemata. At first blush, it appears that schemata are making the search more difficult. For an alphabet with k elements there are only (only?) k^l different strings of length l . Why consider the $(k + 1)^l$ schemata and enlarge the space of concern? Put another way, the length 5 example now has only $2^5 = 32$ different alternative strings. Why make matters more difficult by considering $3^5 = 243$ schemata? In fact, the reasoning discussed in the previous section makes things easier. Do you recall glancing up and down the list of four strings and fitness values and trying to figure out what to do next? We recognized that if we considered the strings separately, then we only had four pieces of information;

however, when we considered the strings, their fitness values, and the similarities among the strings in the population, we admitted a wealth of new information to help direct our search. How much information do we admit by considering the similarities? The answer to this question is related to the number of unique schemata contained in the population. To count this quantity exactly requires knowledge of the strings in a particular population. To get a bound on the number of schemata in a particular population, we first count the number of schemata contained in an individual string, and then we get an upper bound on the total number of schemata in the population.

To see this, consider a single string of length 5: 11111, for example. This string is a member of 2^5 schemata because each position may take on its actual value or a don't care symbol. In general, a particular string contains 2^l schemata. As a result, a population of size n contains somewhere between 2^l and $n \cdot 2^l$ schemata, depending upon the population diversity. This fact verifies our earlier intuition. The original motivation for considering important similarities was to get more information to help guide our search. The counting argument shows that a wealth of information about important similarities is indeed contained in even moderately sized populations. We will examine how genetic algorithms effectively exploit this information. At this juncture, some parallel processing appears to be needed if we are to make use of all this information in a timely fashion.

These counting arguments are well and good, but where does this all lead? More pointedly, of the 2^l to $n \cdot 2^l$ schemata contained in a population, how many are actually processed in a useful manner by the genetic algorithm? To obtain the answer to this question, we consider the effect of reproduction, crossover, and mutation on the growth or decay of important schemata from generation to generation. The effect of reproduction on a particular schema is easy to determine; since more highly fit strings have higher probabilities of selection, on average we give an ever increasing number of samples to the observed best similarity patterns (this is a good thing to do, as is shown in the next chapter); however, reproduction alone samples no new points in the space. What then happens to a particular schema when crossover is introduced? Crossover leaves a schema unscathed if it does not cut the schema, but it may disrupt a schema when it does. For example, consider the two schemata 1***0 and **11*. The first is likely to be disrupted by crossover, whereas the second is relatively unlikely to be destroyed. As a result, schemata of short defining length are left alone by crossover and reproduced at a good sampling rate by reproduction operator. Mutation at normal, low rates does not disrupt a particular schema very frequently and we are left with a startling conclusion. Highly fit, short-defining-length schemata (we call them *building blocks*) are propagated generation to generation by giving exponentially increasing samples to the observed best; all this goes in parallel with no special bookkeeping or special memory other than our population of n strings. In the next chapter we will count how many schemata are processed usefully in each generation. It turns out that the number is something like n^4 . This compares favorably with the number of function evaluations (n). Because this processing leverage is so important (and apparently unique to genetic algorithms), we give it a special name, *implicit parallelism*.

LEARNING THE LINGO

The power behind the simple operations of our genetic algorithm is at least intuitively clearer if we think of building blocks. Some questions remain: How do we know that building blocks lead to improvement? Why is it a near optimal strategy to give exponentially increasing samples to the best? How can we calculate the number of schemata usefully processed by the genetic algorithm? These questions are answered fully in the next chapter, but first we need to master the terminology used by researchers who work with genetic algorithms. Because genetic algorithms are rooted in both natural genetics and computer science, the terminology used in the GA literature is an unholy mix of the natural and the artificial. Until now we have focused on the artificial side of the genetic algorithm's ancestry and talked about strings, alphabets, string positions, and the like. We review the correspondence between these terms and their natural counterparts to connect with the growing GA literature and also to permit our own occasional slip of a natural utterance or two.

Roughly speaking, the *strings* of artificial genetic systems are analogous to *chromosomes* in biological systems. In natural systems, one or more chromosomes combine to form the total genetic prescription for the construction and operation of some organism. In natural systems the total genetic package is called the *genotype*. In artificial genetic systems the total package of strings is called a *structure* (in the early chapters of this book, the structure will consist of a single string, so the text refers to strings and structures interchangeably until it is necessary to differentiate between them). In natural systems, the organism formed by the interaction of the total genetic package with its environment is called the *phenotype*. In artificial genetic systems, the structures decode to form a particular *parameter set*, *solution alternative*, or *point* (in the solution space). The designer of an artificial genetic system has a variety of alternatives for coding both numeric and nonnumeric parameters. We will confront codings and coding principles in later chapters; for now, we stick to our consideration of GA and natural terminology.

In natural terminology, we say that chromosomes are composed of *genes*, which may take on some number of values called *alleles*. In genetics, the position of a gene (its *locus*) is identified separately from the gene's function. Thus, we can talk of a particular gene, for example an animal's eye color gene, its locus, position 10, and its allele value, blue eyes. In artificial genetic search we say that strings are composed of *features* or *detectors*, which take on different *values*. Features may be located at different *positions* on the string. The correspondence between natural and artificial terminology is summarized in Table 1.3.

Thus far, we have not distinguished between a gene (a particular character) and its locus (its position); the position of a bit in a string has determined its meaning (how it decodes) uniformly throughout a population and throughout time. For example, the string 10000 is decoded as a binary unsigned integer 16 (base 10) because implicitly the 1 is in the 16's place. It is not necessary to limit codings like this, however. A later chapter presents more advanced structures that treat locus and gene separately.

TABLE 1.3 Comparison of Natural and GA Terminology

Natural	Genetic Algorithm
chromosome	string
gene	feature, character, or detector
allele	feature value
locus	string position
genotype	structure
phenotype	parameter set, alternative solution, a decoded structure
epistasis	nonlinearity

SUMMARY

This chapter has laid the foundation for understanding genetic algorithms, their mechanics and their power. We are led to these methods by our search for robustness; natural systems are robust—efficient and efficacious—as they adapt to a wide variety of environments. By abstracting nature's adaptation algorithm of choice in artificial form we hope to achieve similar breadth of performance. In fact, genetic algorithms have demonstrated their capability in a number of analytical and empirical studies.

The chapter has presented the detailed mechanics of a simple, three-operator genetic algorithm. Genetic algorithms operate on populations of strings, with the string coded to represent some underlying parameter set. Reproduction, cross-over, and mutation are applied to successive string populations to create new string populations. These operators are simplicity itself, involving nothing more complex than random number generation, string copying, and partial string exchanging; yet, despite their simplicity, the resulting search performance is wide-ranging and impressive. Genetic algorithms realize an innovative notion exchange among strings and thus connect to our own ideas of human search or discovery. A simulation of one generation of the simple genetic algorithm has helped illustrate both the detail and the power of the method.

Four differences separate genetic algorithms from more conventional optimization techniques:

1. Direct manipulation of a coding
2. Search from a population, not a single point
3. Search via sampling, a blind search
4. Search using stochastic operators, not deterministic rules

Genetic algorithms manipulate decision or control variable representations at the string level to exploit similarities among high-performance strings. Other methods usually deal with functions and their control variables directly. Because

genetic algorithms operate at the coding level, they are difficult to fool even when the function may be difficult for traditional schemes.

Genetic algorithms work from a population; many other methods work from a single point. In this way, GAs find safety in numbers. By maintaining a population of well-adapted sample points, the probability of reaching a false peak is reduced.

Genetic algorithms achieve much of their breadth by ignoring information except that concerning payoff. Other methods rely heavily on such information, and in problems where the necessary information is not available or difficult to obtain, these other techniques break down. GAs remain general by exploiting information available in any search problem. Genetic algorithms process similarities in the underlying coding together with information ranking the structures according to their survival capability in the current environment. By exploiting such widely available information, GAs may be applied in virtually any problem.

The transition rules of genetic algorithms are stochastic; many other methods have deterministic transition rules. A distinction exists, however, between the randomized operators of genetic algorithms and other methods that are simple random walks. Genetic algorithms use random choice to guide a highly exploitative search. This may seem unusual, using chance to achieve directed results (the best points), but nature is full of precedent.

We have started a more rigorous appraisal of genetic algorithm performance through the concept of schemata or similarity templates. A schema is a string over an extended alphabet, $\{0,1,*\}$ where the 0 and the 1 retain their normal meaning and the * is a wild card or don't care symbol. This notational device greatly simplifies the analysis of the genetic algorithm method because it explicitly recognizes all the possible similarities in a population of strings. We have discussed how building blocks—short, high-performance schemata—are combined to form strings with expected higher performance. This occurs because building blocks are sampled at near optimal rates and recombined via crossover. Mutation has little effect on these building blocks; like an insurance policy, it helps prevent the irrecoverable loss of potentially important genetic material.

The simple genetic algorithm studied in this chapter has much to recommend it. In the next chapter, we will analyze its operation more carefully. Following this, we will implement the simple GA in a short computer program and examine some applications in practical problems.

■ PROBLEMS

1.1. Consider a black box containing eight multiple-position switches. Switches 1 and 2 may be set in any of 16 positions. Switches 3, 4, and 5 are four-position switches, and switches 6–8 have only two positions. Calculate the number of unique switch settings possible for this black box device.

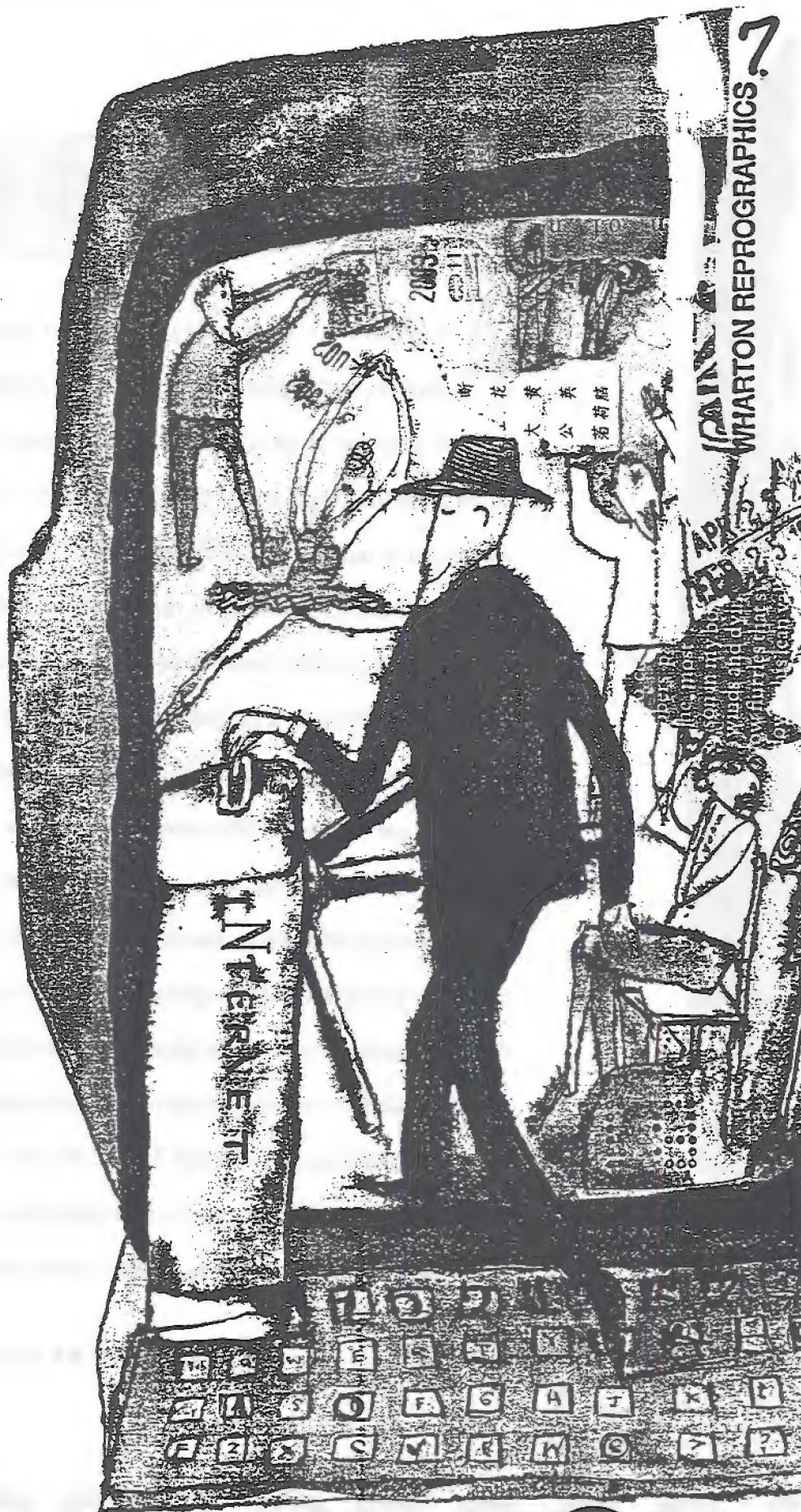
1.2. For the black box device of Problem 1.1, design a natural string coding that uses eight positions, one position for each switch. Count the number of switch

The first of these is the fact that the system is not a simple one. It is a complex system, and the complexity is not only in the number of components, but also in the way they are connected. The second is that the system is not a static one. It is a dynamic system, and the dynamics are not only in the way the components interact, but also in the way the system evolves over time. The third is that the system is not a linear one. It is a non-linear system, and the non-linearity is not only in the way the components interact, but also in the way the system evolves over time. The fourth is that the system is not a deterministic one. It is a stochastic system, and the stochasticity is not only in the way the components interact, but also in the way the system evolves over time. The fifth is that the system is not a simple one. It is a complex system, and the complexity is not only in the number of components, but also in the way they are connected. The sixth is that the system is not a static one. It is a dynamic system, and the dynamics are not only in the way the components interact, but also in the way the system evolves over time. The seventh is that the system is not a linear one. It is a non-linear system, and the non-linearity is not only in the way the components interact, but also in the way the system evolves over time. The eighth is that the system is not a deterministic one. It is a stochastic system, and the stochasticity is not only in the way the components interact, but also in the way the system evolves over time.

APPENDIX 2

The first of these is the fact that the system is not a simple one. It is a complex system, and the complexity is not only in the number of components, but also in the way they are connected. The second is that the system is not a static one. It is a dynamic system, and the dynamics are not only in the way the components interact, but also in the way the system evolves over time. The third is that the system is not a linear one. It is a non-linear system, and the non-linearity is not only in the way the components interact, but also in the way the system evolves over time. The fourth is that the system is not a deterministic one. It is a stochastic system, and the stochasticity is not only in the way the components interact, but also in the way the system evolves over time.

With its growing ease of use and burgeoning popularity, the Internet is fast becoming the all-purpose information superhighway we've been hearing so many promises about. But can it survive the transition from a government protectorate to a free-market medium?



< INTERNET @ CF



BY HERB BRODY

AS WITH ANY ROMANCE, the world's present infatuation with the Internet has emphasized the magic and tended to ignore the practical. Couples falling in love dote on each other's wonderful traits. It is only a little later that they float down from the clouds and tackle the economics—how big a house they can afford, what kind of income they will need. □ The Internet, for most people, is still a new love. Ignore for a moment the few tens of thousands of people who inhabited the Net before the phenomenal population boom began in the late 1980s. For the rest of us, the Internet is still exciting and not a little bit mysterious. □ But the oddity is that nobody seems to be paying for all the informational goodies that can pour into our computers like water from a broken pipe. You might pay a few dollars a month for the privilege of being connected, but once you slide past the electronic turnstile, it's an all-you-can-consume buffet of bits, from plain vanilla e-mail to participation in online discussion groups to searches of Library of Congress databases to the latest satellite images of the planet. Want to send a 10-page letter to a friend in Australia? Go ahead—no extra charge. Want a nifty piece of software that lets you browse the Net by pointing and clicking? Take it, it's free. Want to mail a fund-raising appeal to 10,000 people? The Internet converts this from a \$3,200 postal endeavor into one that's more or less on the house. Internet users seem to have found a kind of surreal

ILLUSTRATIONS BY BLAIR THORNLEY

ROSSROADS. \$\$\$ \$ >

restaurant where they can order a bottomless cup of coffee or a lobster dinner for 100 friends and no one ever presents an itemized bill.

Part of the reason is that, at least until the last few years, most members of the Internet Nation plugged in through computers at their workplace or university, so costs were a kind of invisible overhead that someone else worried about. At MIT, for example, a high-speed fiber-optic network called MITnet links computers all over campus. One of these computers serves as a "gateway" that connects MITnet to one of 17 regional networks, in this case called Nearnnet and operated by Bolt Beranek and Newman, a Cambridge-based technology consulting company. Nearnnet, in turn, is connected to a "backbone" known as NSFNet because it is funded by the National Science Foundation. NSFNet itself is operated by ANS, a not-for-profit company that has leased high-capacity fiber-optic telephone lines from the same companies that handle long-distance telephone traffic—AT&T, MCI, and Sprint. Each organization pays a flat rate to the broader system it taps into; individual users are essentially insulated from cost burdens regardless of the volume of their use.

With similar hierarchical connections, commercial on-line services such as Prodigy and America Online give individual subscribers e-mail privileges on the Internet as well as access to some of its more popular resources, such as the Usenet newsgroups (online bulletin boards on hundreds of topics). These commercial services are heavily promoting such connections, particularly the ability of subscribers to tap into the World Wide Web, an interwoven collection of Internet resources that allows point-and-click navigation without mastery of arcane commands. But although this access brings entrance into an electronic universe where interactivity is not just a marketing slogan but a way of life, the cost is usually just a fraction of what users pay for cable television.

That situation may change as the Internet detaches from the government umbilical cord that has nurtured it through its infancy. Beginning April 30 of this year, the NSF will no longer pay to operate the backbone net-

work. The portion of NSF funding that goes to the 17 regionals is also now on a five-year "sunset schedule," dropping gradually to zero by fiscal 1998.

Under the new arrangement, the federal government will grant this dwindling amount of money directly to each regional network and instruct it to shop for backbone service on the private market. The transition resembles, in one sense, the breakup of the Bell Sys-

tem a decade ago. But the government is not stepping out of the Internet picture altogether. NSF is setting up and funding three "network access points," or NAPs; any company that wants to operate an Internet backbone must connect to each of these three NAPs, which are to be located in New Jersey, Chicago, and California. To qualify as a backbone service provider, a company must agree to accept Internet transmissions that arrive at each NAP from every other backbone company.

The withdrawal of a large part of government support will not by itself significantly raise prices for users. NSF's total funding for the Internet is only about \$20 million a year. The companies, universities, and individuals that use the Net pay many times that amount, and dividing that \$20 million over the number of present users yields only about \$1 per person per year. As the Net grows in popularity, that burden may diminish further. What worries some analysts, however, is that the nature of information being sent over the Internet is changing rapidly, with potential implications for the system's cost and ease of access.

DIGITAL CONGESTION

Until about two years ago, the overwhelming majority of Net users were transmitting simple text such as e-mail messages and Usenet postings. Text is a highly efficient method of communication: the words composing a page of the *Encyclopedia Britannica*, for instance, can be encoded in standard ASCII form using fewer than 10 kilobytes. But new software and more powerful desktop computers have made it practical to send high-resolution color images, sound files, even full-motion video—anyone with a camcorder and

*Millions of people
enjoy low-cost access
to the Internet's information cornucopia.
Now, without the
federal subsidies that
have built and sustained
it, the Net will have to
make it on its own—
forcing decisions on
whether, and how much,
users might pay.*

HERB BRODY is a senior editor of Technology Review. His Internet address is hbrody@mit.edu.

a multimedia computer can conduct a videoconference, for example, over the Net. Such uses consume orders of magnitude more capacity, or "bandwidth," than text. By swamping the network with video signals, relatively few users can temporarily overload portions of the Net.

The World Wide Web has the potential to exacerbate this problem, since Web browsers can easily, almost inadvertently, trigger the transmission of huge amounts of data. Wave a mouse

around the screen, click once on an appealing picture, and megabytes start flowing. Without the Web, users must type in a command to retrieve that information—a step that can at least give them pause.

To understand both the Net's capacity to transmit such information and its vulnerability to overload, compare it with the familiar telephone system. That network uses a technique known as circuit switching: when you dial your grandmother's number, you are instructing the system's switches to establish a connection between your telephone and hers. This connection is maintained for the duration of the call: as long as you are on the line, no one else can use this circuit. You are consuming a scarce resource and pay for the privilege.

Although most Internet traffic physically flows through the telephone wires, information is packaged and routed much differently. Each transmission is broken up into discrete "packets" containing roughly 200 bytes (packet size varies). Each packet is stamped with the recipient's address. The packets then bounce from computer to computer along the Net, each computer examining the address and deciding where to send them next for the most efficient transmission. Since these decisions depend on conditions at the moment, the packets may travel different routes to reach the same destination. Eventually all the packets arrive at the receiving computer, which reassembles them into the original form.

This structure—or architecture, as computer sci-

tists like to call it—stems in part from the Internet's origins as a defense project. A packet-switching system is difficult to eavesdrop on, since messages are scattered to the electronic winds before finally coalescing at the receiving point. The design also lowers the risk that a military attack would disrupt communications—a primary concern in the 1950s and '60s when ARPAnet, the Internet's ancestor, was designed by the Pentagon's

Advanced Research Projects Agency. The reasoning was brutally straightforward: if an enemy attack were to knock out the Washington-to-New York connection, say, information would still move between these two cities, albeit in a roundabout manner. A circuit-switched network like the telephone system offers only limited flexibility in this regard because a circuit must be established before communication begins; a packet-switched network can dynamically "heal" itself in mid-transmission.

Although motivated initially by security concerns, packet-switching technology has profound implications for the economics of network communi-

cations. When someone sends something over the Internet—say, a piece of e-mail—the packets do not consume a scarce resource in the same way that a phone call does. If a router is busy, incoming packets simply queue up and wait their turn. Longer lines translate into delays, not busy signals. For the uses of the Internet that have prevailed so far, such lags don't make much difference. Unlike telephone conversations, which take place in real time, e-mail communications can easily tolerate delays of many seconds or even minutes. However, the advent of multimedia services on the Internet is making delays less tolerable. If packets queue up at a router, quality of service can deteriorate; video appears jumpy, for example, and moving from one World Wide Web link to another can take so long that the medium becomes more an annoyance than an adventure.

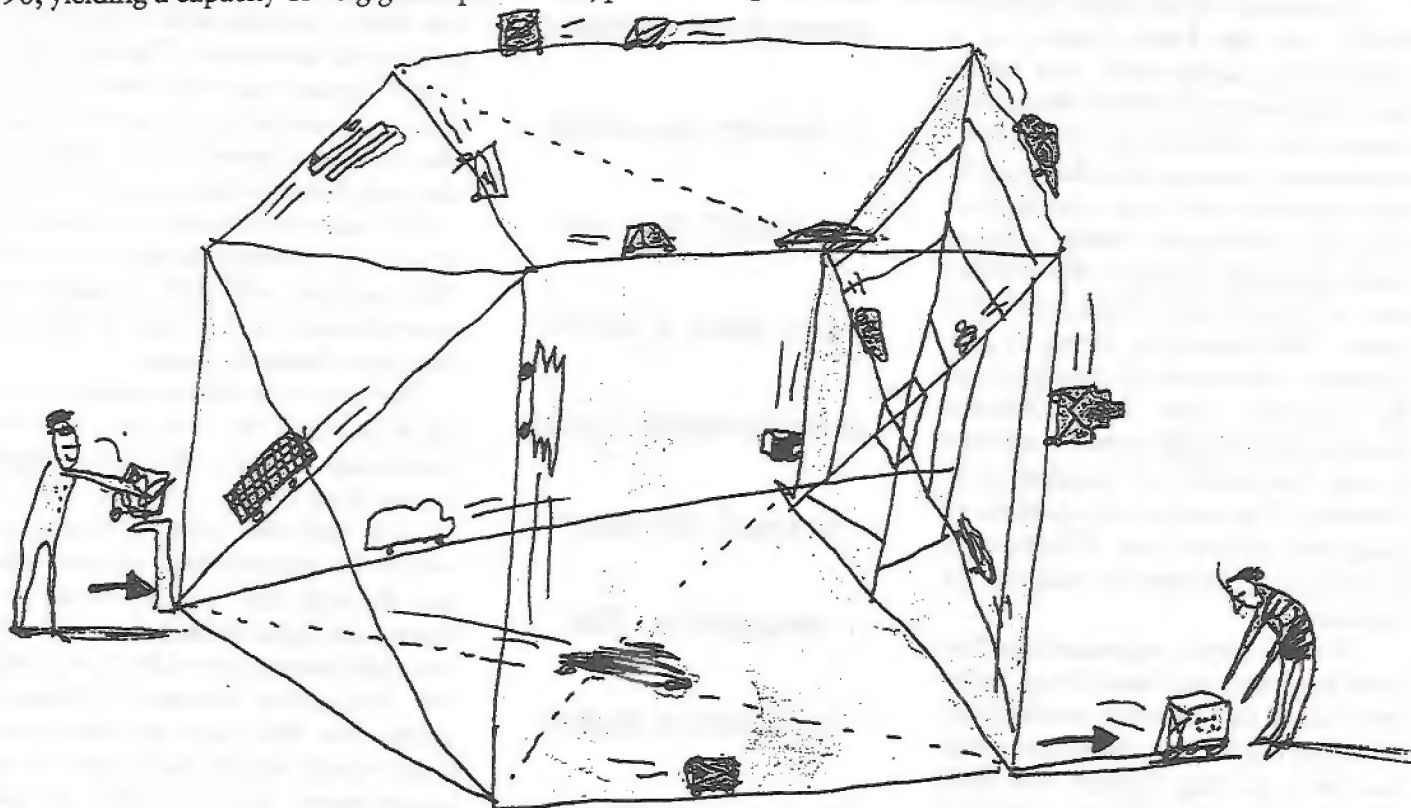
It is possible that advances in technology will pro-



vide the needed capacity. The best fibers used in the long-distance links that carry both voice and data traffic can accommodate 2.4 gigabits (billion bits) per second. Over the next two years, upgrades to optical transmitters and receivers will quadruple that data rate. Even better, after more than a decade of development, telecommunications engineers are perfecting a system called wavelength division multiplexing, which enables a single fiber to carry multiple channels of information, each encoded on laser light of a slightly differing wavelength. Such multiplexed systems will be in place by 1998, yielding a capacity of 40 gigabits per second, pre-

as videoconferencing, such delays become intolerable.

Congestion on the Internet is already hampering attempts to use it for new applications during peak business hours, says Jeffrey Mackey-Mason, an economist at the University of Michigan. The problem becomes particularly acute when some special event occurs. After the comet Shoemaker-Levy struck Jupiter, for example, and people downloaded the dramatic telescope images, large portions of the Internet slowed down. In such situations, urgent transmissions, such as a potentially lifesaving videoconference between a surgeon and a radiologist, might queue up behind a home movie that



dicts Vinton Cerf, a senior vice-president at MCI's data services division and president of the Internet Society, a nonprofit organization that promotes Internet usage and standardization. These radical leaps in performance are coming as costs of all component technologies—fibers, lasers, and electronics—decline. Technology, in other words, has the capacity to abolish near-term bandwidth scarcity.

Still, if the recent past is any guide, the demand for bandwidth will grow at least as fast as the supply. According to University of Michigan economist Hal Varian, while NSFnet now operates at only 5 percent of capacity, the volume of packet flow is rising by 6 percent per month. At that rate, average traffic volume will reach 20 percent of capacity in only two years. During times of peak use, however, the amount of information put onto the Net far exceeds this 20 percent average, and packets that don't "fit" have to wait until a channel is clear. As the Internet is used more and more for real-time forms of communication such

someone put on the Net just for fun. In effect, the Net can be dominated by people with a lot of time on their hands, and there is no provision for buying one's way to the front of the line.

TO CHARGE OR NOT TO CHARGE

Some analysts therefore contend that the nation needs some kind of disincentive to unbridled consumption of the Net's capacity. If people have to pay for what they do, they will tend to do less, says Padmanabhan Srinagesh, an engineer at Bell Communications Laboratories, or Bellcore. Net users, in other words, might have to say goodbye to the freedom of a flat rate. The Internet would instead be metered, with users paying by the message, by the byte, or by the Web page, just as they now pay by the kilowatt-hour for electricity or by the minute for long-distance phone calls.

Philip Gross, vice-president for Internet engineering at MCI's data services division, agrees that some sort of

usage-based pricing is "inevitable," if only so the multiple companies handling backbone traffic can count packets and settle accounts with one another. But the solution to the problem is not as simple as counting packets or bytes. Take a typical transaction: the transfer of a large software file. User A sends a 100-byte request to User B, who responds by transmitting a 1-megabyte program. A naive billing system would charge User B 10,000 times more than User A, even though User A initiated the transaction and received all of its benefits.

Opponents of fees argue that they would rob the Net's vitality as a workplace, playground, and social club. The electronic culture that extols instant and perpetual electronic connectedness presupposes that people will not have to fret over a ticking fare meter in cyberspace. Usage pricing might especially dampen the enthusiasm of school and home computer users. "We should be trying to preserve the motivation of people to use the Internet," says David Wasley, director of computer network services at the University of California at Berkeley. The recent introduction of usage fees in Australia, Wasley says, led to a rapid decline in e-mail sent by students.

Not-for-profit organizations that have begun to rely heavily on Internet mailing lists to reach activists and potential donors are particularly vulnerable to pricing changes. Any kind of per-use charge will have a chilling effect on some of these fledgling exercises in "electronic democracy," asserts James Love, president of the Washington-based Taxpayers' Assets Project, a group that monitors the outcome of privatization efforts. "Say you send a message a day to everyone on a 10,000-name list," he says. "If you have to pay per transaction, that adds up."

Other complications arise as well. Part of the Internet's value—and charm—lies in its utter transcendence of geography. In today's system, a Net surfer who downloads pictures of Jupiter need not care whether the computer holding these images is 10 miles or 10,000 miles away. All information stored in any computer on the network is, in effect, stored everywhere. Because the Net has traditionally rendered geographical distance a secondary concern, users often don't even know

exactly "where" they are traveling. "If you try to charge based on distance or on the number of bits, then the Internet falls apart," says Edward Krol of the University of Illinois, author of *The Whole Internet User's Guide & Catalog*. "If the best resource happens to be in Belgium, you just use it."

Fortunately, the present flat-rate pricing does have some built-in protection against network congestion. As it now stands, the cost of Net access varies with the bandwidth of the basic connection. A user operating from a home or office with a 9,600-bit-per-second link pays less than a business with a 56-kilobit-per-second connection. The size of this "pipe" limits usage just the way the size of a house's electrical service limits the amount of power it can draw off the grid. Nevertheless, when millions of Net users raise their expectations of what is possible on the network, "the Internet will melt," maintains Scott Shenker, a researcher at Xerox's Palo Alto Research Center.

One way to avoid any potential crisis is to outfit the Net with mechanisms that perform electronic triage. Usage that cannot tolerate delays, such as real-time videoconferencing, would be stamped high priority and sail through the Net like an ambulance with siren wailing. Users would pay their Internet provider a premium for this special treatment. Shenker points out that such priority-based preferences would lead logically to usage-based pricing, since people would think twice about conducting real-time videoconferences if they had to pay dearly for the privilege of displacing so much other activity. Most users would routinely put a lowest-priority tag on e-mail and text postings to newsgroups, which rarely require rapid delivery, and the fees paid by senders who demand high priority would subsidize any cost associ-

ated with such transmission.

Although companies that provide Net access have stuck mostly to flat-fee structures, they may start promoting congestion pricing as a marketing edge as the number of providers proliferates. Many users will presumably appreciate the ability to pay for what amounts to a guarantee of immediate transmission.

One barrier to such a move is that today's network has no means of tracking packets sent and received. All

*Information sent
through the Internet
is broken into small
"packets" that can
travel along a variety
of independent paths
to reach the same
destination. This
architecture makes
the Internet efficient
but still vulnerable
to overload.*

Internet communication is governed by a set of rules, or "protocols," called TCP/IP (transmission control protocol/Internet protocol). TCP/IP specifies the method by which any digital data—whether they represent text, graphics, or anything else—are transmitted, routed, checked for errors, and, if need be, resent. But nothing in the protocols provides the detailed information that commercial telecommunications companies need to provide a billing record, says MCI's Gross.

Even if the protocols did gather such information, the accounting process is bound to be expensive. More than half of what customers pay for a telephone call goes to cover the cost of the accounting system, contends Wasley. Thus any attempt to bill for Internet use could become a case of self-fulfilling prophecy: the very act of collecting the necessary information could raise the network's operating cost to the point that users will have to pay more. Anthony Rutkowski, executive director of the Internet Society and a former vice-president at Sprint Telecommunications, thinks "it won't be worth the trouble to account for users' consumption of the network's capacity."

Whether and how to devise a workable billing system if and when usage-based pricing arrives is a decision that will have to be made cooperatively by the companies that carry Internet traffic, along with the Internet

Engineering Task Force—an organization with representatives from government, research and educational institutions, and vendors from all over the world that writes technical standards for the Internet.

A CHANGING PICTURE

The next few years will be a time of shakeout in the burgeoning Internet business. Because the Internet operates as a loose cluster of networks, no one is really "in charge," and each provider of local, regional, and backbone service is free to price Internet access any way it chooses—each, of course, influenced by the price charged by *its* access supplier.

The most likely short-term scenario is for flat-rate service to continue as the norm. In fact, the prevailing assumption is that the smartest way to do business will be to maintain the status quo. Online society has mushroomed on the basis of unlimited use, a structure that encourages a kind of freewheeling exploration. The companies that sell Internet access understand this appeal and seem loath to tamper with it. "We don't want to kill the goose that laid the golden egg," says MCI's Gross. He insists that MCI "will not unilaterally impose" usage-based pricing. "We're very content to operate in the current Internet mode" in the near term, he

says, and expects to make no major changes in the next year or two. Eric Aupperle, president of Merit Network, the regional network serving Michigan, echoes this sentiment. "My sense is that the community wants flat-fee access," he says, "and that's how it's going to be."

One compromise pricing method is for an Internet customer to declare at the outset whether it will be a high-volume or low-volume user. For a given bandwidth, the Internet service provider could charge the low-volume user less than a high-volume user. That way a small company can get the benefit of an affordable, high-capacity access. MCI is looking to put something like this in place "in the very short term," says Gross.

The amount of change that end users will experience depends on the type of Internet service they have become accustomed to. Clients of Near-net in the Northeast and of Barnet in California will feel barely a ripple during this transition. Those networks have been accepting commercial business for years, and so have relied less on NSF. But in other areas of the country, the regional networks have continued to depend heavily on NSF for funding. There, prices to universities and other organizations that hook into the regional network will probably rise.

INTO THE MARKETPLACE

As federal funding for the Internet winds down, following it into extinction are the NSF's "acceptable use policies," which have restricted use of the Net for profit-making activity. As the Internet is reborn as a medium not just of public discourse but of commercial opportunity, the revenues generated by such activity might well render moot the questions of per-use charges and ensure access by nonprofit groups. Just as retail stores pay rent to occupy a shopping mall, companies that make a profit on the Net will pay telecommunications companies to be there. Microsoft is heading in this general direction with its plan to offer Internet access through Windows 95, the long-awaited version of its popular Windows software for personal computers. Microsoft will obtain its revenue not from consumers, who will pay the company only a nominal fee for Internet access, but from businesses that set up shop on the Net.

Commerce on the Internet is still embryonic. A few companies publish online catalogs, but transactions are still consummated by telephone and credit card and the product arrives in a UPS truck. But it is in the marketing of information-based products that the Internet's business potential can be more fully tapped. Software.Net, for example, delivers computer programs through the Net. Other companies are similarly gearing up to distribute music and digitized art online, according to Bob O'Keefe, a professor at Rensselaer Polytechnic Institute's School of Management. Commercial sponsorship also made possible "free" radio and television broadcast; following this model, advertising revenue could pay for on-line publications. Such advertising is potentially of more value than printed or broadcast ads to consumers, who can obtain precisely the product information they need with a few mouse clicks on an unobtrusive icon.

Although such moves may solve some pricing problems, the biggest hurdles to Internet access for many people are not fees for service but the cost of a computer. Commercial activity on the Net could help here, too. As a marketplace, the Internet can be subject to a process as old as the Net is modern: taxation. One suggestion, from David Farber of the University of Pennsylvania, is to impose a 10 percent sales tax on business conducted over the Internet. This revenue could create a pool of funds to buy Net terminals that would

be distributed widely at libraries and community centers. Given the resistance to new taxes that now dominates the political landscape, however, such a scheme seems far-fetched.

While equity of access is far from guaranteed in the near term, the public should in the long run benefit as the Internet is released from the simultaneously nurturing and smothering federal sponsorship. Decades of government support have built a communications infrastructure that fosters experimentation. For the last eight or nine years, the NSF has been in the business of "market building," says NSFNet program officer David Staube—constructing an infrastructure and trying to persuade institutions and individuals to use it. That phase has passed. Now, he says, "the market can stand on its own—without our seed money." ■

*Usage-based
fees could defray
operating costs and
possibly relieve con-
gestion and delays.
But the idea of a
ticking fare meter
is anathema to
Net culture.*

TO VIEW THIS ARTICLE WITH INTERACTIVE LINKS, VISIT OUR WORLD WIDE WEB SERVER AT
< [HTTP://WEB.MIT.EDU/TECHREVIEW/WWW/](http://web.mit.edu/techreview/www/) >

Edgar H. Sibley
Panel Editor

An evaluation of a large, operational full-text document-retrieval system (containing roughly 350,000 pages of text) shows the system to be retrieving less than 20 percent of the documents relevant to a particular search. The findings are discussed in terms of the theory and practice of full-text document retrieval.

AN EVALUATION OF RETRIEVAL EFFECTIVENESS FOR A FULL-TEXT DOCUMENT-RETRIEVAL SYSTEM

DAVID C. BLAIR and M. E. MARON

Document retrieval is the problem of finding stored documents that contain useful information. There exist a set of documents on a range of topics, written by different authors, at different times, and at varying levels of depth, detail, clarity, and precision, and a set of individuals who, at different times and for different reasons, search for recorded information that may be contained in some of the documents in this set. In each instance in which an individual seeks information, he or she will find some documents of the set useful and other documents not useful: the documents found useful are, we say, *relevant*; the others, not relevant.

How should a collection of documents be organized so that a person can find all and only the relevant items? One answer is automatic full-text retrieval, which on its surface is disarmingly simple: Store the full text of all documents in the collection on a computer so that every character of every word in every sentence of every document can be located by the machine. Then, when a person wants information from that stored collection, the computer is instructed to search for all documents containing certain specified words and word combinations, which the user has specified.

Two elements make the idea of automatic full-text retrieval even more attractive. On the one hand, digital technology continues to provide computers that are larger, faster, cheaper, more reliable, and easier to use; and, on the other hand, full-text retrieval avoids the

need for human indexers whose employment is increasingly costly and whose work often appears inconsistent and less than fully effective.

A pioneering test to evaluate the feasibility of full-text search and retrieval was conducted by Don Swanson and reported in *Science* in 1960 [6]. Swanson concluded that text searching by computer was significantly better than conventional retrieval using human subject indexing. Ten years later, in 1970, Salton, also in *Science*, reported optimistically on a series of experiments on automatic full-text searching [3].

This paper describes a large-scale, full-text search and retrieval experiment aimed at evaluating the effectiveness of full-text retrieval. For the purposes of our study, we examined IBM's full-text retrieval system, STAIRS. STAIRS, an acronym for "STorage And Information Retrieval System," is a very fast, large-capacity, full-text document-retrieval system. Our empirical study of STAIRS in a litigation support situation showed its retrieval effectiveness to be surprisingly poor. We offer theoretical reasons to explain why this poor performance should not be surprising and also why our experimental results are not inconsistent with the earlier more favorable results cited above. The retrieval problems we describe would be problems with any large-scale, full-text retrieval system, and in this sense our study should not be seen as a critique of STAIRS alone, but rather a critique of the principles on which it and other full-text document-retrieval systems are based.

THE ALLURE OF FULL-TEXT DOCUMENT RETRIEVAL

Retrieving document texts by subject content occupies a special place in the province of information retrieval because, unlike data retrieval, the richness and flexibility of natural language have a significant impact on the conduct of a search. The indexer chooses subject terms that will describe the informational content of the documents included in the database, and the user describes his or her information need in terms of the subject descriptors actually assigned to the documents (Figure 1). However, there are no clear and precise rules to govern the indexers' choice of appropriate subject terms, so that even trained indexers may be inconsistent in their application of subject terms. Experimental studies have demonstrated that different indexers will generally index the same document differently [9], and even the same individual will not always select the identical index terms if asked at a later time to index a document he or she has already indexed. The problems associated with manual assignment of subject descriptors make computerized, full-text document retrieval extremely appealing. By entering the entire, or the most significant part of, a document text onto the database, one is freed, it is argued, from the inherent evils of manually creating document records reflecting the subject content of a particular document; among these, the construction of an indexing vocabulary, the train-

ing of indexers, and the time consumed in scanning/reading documents and assigning context and subject terms. The economies of full-text search are appealing, but for it to be worthwhile, it must also provide satisfactory levels of retrieval effectiveness.

MEASURING RETRIEVAL EFFECTIVENESS

Two of the most widely used measures of document-retrieval effectiveness are Recall and Precision. Recall measures how well a system retrieves all the relevant documents; and Precision, how well the system retrieves only the relevant documents. For the purposes of this study, we define a document as relevant if it is judged useful by the user who initiated the search. If not, then it is nonrelevant (see [4]). More precisely, Recall is the proportion of relevant documents that the system retrieves, the ratio of x/n_2 (Figure 2). Notice that one can interpret Recall as the probability that a relevant document will be retrieved. Precision, on the other hand, measures how well a system retrieves only the relevant documents; it is defined as the ratio x/n_1 and can be interpreted as the probability that a retrieved document will be relevant.

THE TEST ENVIRONMENT

The database examined in this study consisted of just under 40,000 documents, representing roughly 350,000 pages of hard-copy text, which were to be used in the

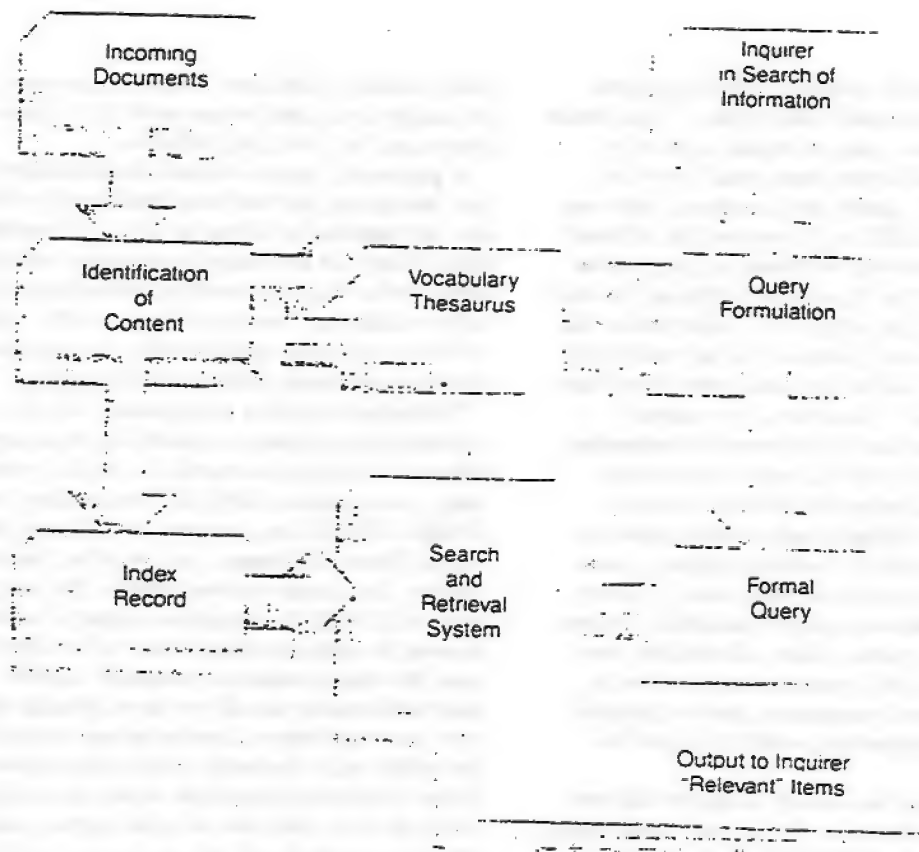


FIGURE 1. The Dynamics of Information Retrieval

$$\text{Recall} = \frac{\text{Number of Relevant and Retrieved}}{\text{Total Number Relevant}} = \frac{x}{n_2}$$

(Theoretical Set of All Relevant Documents)

$$\text{Precision} = \frac{\text{Number of Relevant and Retrieved}}{\text{Total Number Retrieved}} = \frac{x}{n_1}$$

FIGURE 2. Definitions of Precision and Recall

defense of a large corporate law suit. Access to the documents was provided by IBM's STAIRS/TLS software (STorage And Information Retrieval System/Thesaurus Linguistic System). STAIRS software represents state-of-the-art software in full-text retrieval. It provides facilities for retrieving text where specified words appear either singly or in complex Boolean combinations. A user can specify the retrieval of text in which words appear together anywhere in the document, within the same paragraph, within the same sentence, or adjacent to each other (as in "New"adjacent "York"). Retrieval can also be performed on fields such as author, date, and document number. STAIRS provides ranking functions that permit the user to order retrieved sets of 200 documents or less in either ascending or descending numerical (e.g., by date) or alphabetic (e.g., by author) order. In addition, retrieved sets of less than 200 documents can also be ordered by the frequency with which specified search terms occur in the retrieved documents. The Thesaurus Linguistic System (TLS) provides the facilities to manually create an interactive thesaurus that can be called up by the user to semantically broaden (or narrow) his or her searches; it allows the designer to specify semantic relationships between search terms such as "narrower than," "broader than," "related to," "synonymous with," as well as automatic phrase decomposition. STAIRS/TLS thus represents a comprehensive full-text document-retrieval system.

THE EXPERIMENTAL PROTOCOL

To test how well STAIRS could be used to retrieve *all* and *only* the documents relevant to a given request for information, we wanted in essence to determine the values of Recall (percentage of relevant documents retrieved) and Precision (percentage of retrieved documents that are relevant). Although Precision is an important measure of retrieval effectiveness, it is meaningless unless compared to the level of Recall desired by the user. In this case, the lawyers who were to use the system for litigation support stipulated that they must be able to retrieve at least 75 percent of all the documents relevant to a given request for information, and that they regarded this entire 75 percent as essential to the defense of the case. (The lawyers divided the relevant retrieved documents into three groups: "vital," "satisfactory," and "marginally relevant." All other retrieved documents were considered "irrelevant.")

CONDUCT OF THE TEST

For the test, we attempted to have the retrieval system used in the same way it would have been during actual litigation. Two lawyers, the principal defense attorneys in the suit, participated in the experiment. They generated a total of 51 different information requests, which were translated into formal queries by either of two paralegals, both of whom were familiar with the case and experienced with the STAIRS system. The paralegals searched on the database until they found a set of documents they believed would satisfy one of the initial requests. The original hard copies of these documents were retrieved from files, and xerox copies were sent to the lawyer who originated the request. The lawyer then evaluated the documents, ranking them according to whether they were "vital," "satisfactory," "marginally relevant," or "irrelevant" to the original request. The lawyer then made an overall judgment concerning the set of documents received, stating whether he or she wanted further refinement of the query and further searching. The reasons for any subsequent query revisions were made in writing and were fully recorded. The information-request and query-formulation procedures were considered complete only when the lawyer stated in writing that he or she was satisfied with the search results for that particular query (i.e., in his or her judgment, more than 75 percent of the "vital," "satisfactory," and "marginally relevant" documents had been retrieved). It was only at this point that the task of measuring Precision and Recall was begun. (A diagram of the information-request procedure is given in Figure 3.) The lawyers and paralegals were permitted as much interaction as they thought necessary to ensure highly effective retrieval. The paralegals were able to seek clarification of the lawyers' information request in as much detail and as often as they desired, and the lawyers were encouraged to continue requesting information from the database until they were satisfied they had enough information to defend the lawsuit on that particular issue or query. In the test, each query required a number of revisions, and the lawyers were not generally satisfied until many retrieved sets of documents had been generated and evaluated.

Precision was calculated by dividing the total number of relevant (i.e., "vital," "satisfactory," and "marginally relevant") documents retrieved by the total number of retrieved documents. If two or more retrieved sets were generated before the lawyer was satisfied with the results of the search, then the retrieved set considered for calculating Precision was computed as the *union* of all retrieved sets generated for that request. (Documents that appeared in more than one retrieved set were automatically excluded from all but one set.)

Recall was considerably more difficult to calculate since it required finding relevant documents that had not been retrieved in the course of the lawyers' search. To find the *unretrieved* relevant documents, we developed sample frames consisting of subsets of the unretrieved database that we believed to be rich in relevant documents (and from which duplicates of retrieved rel-

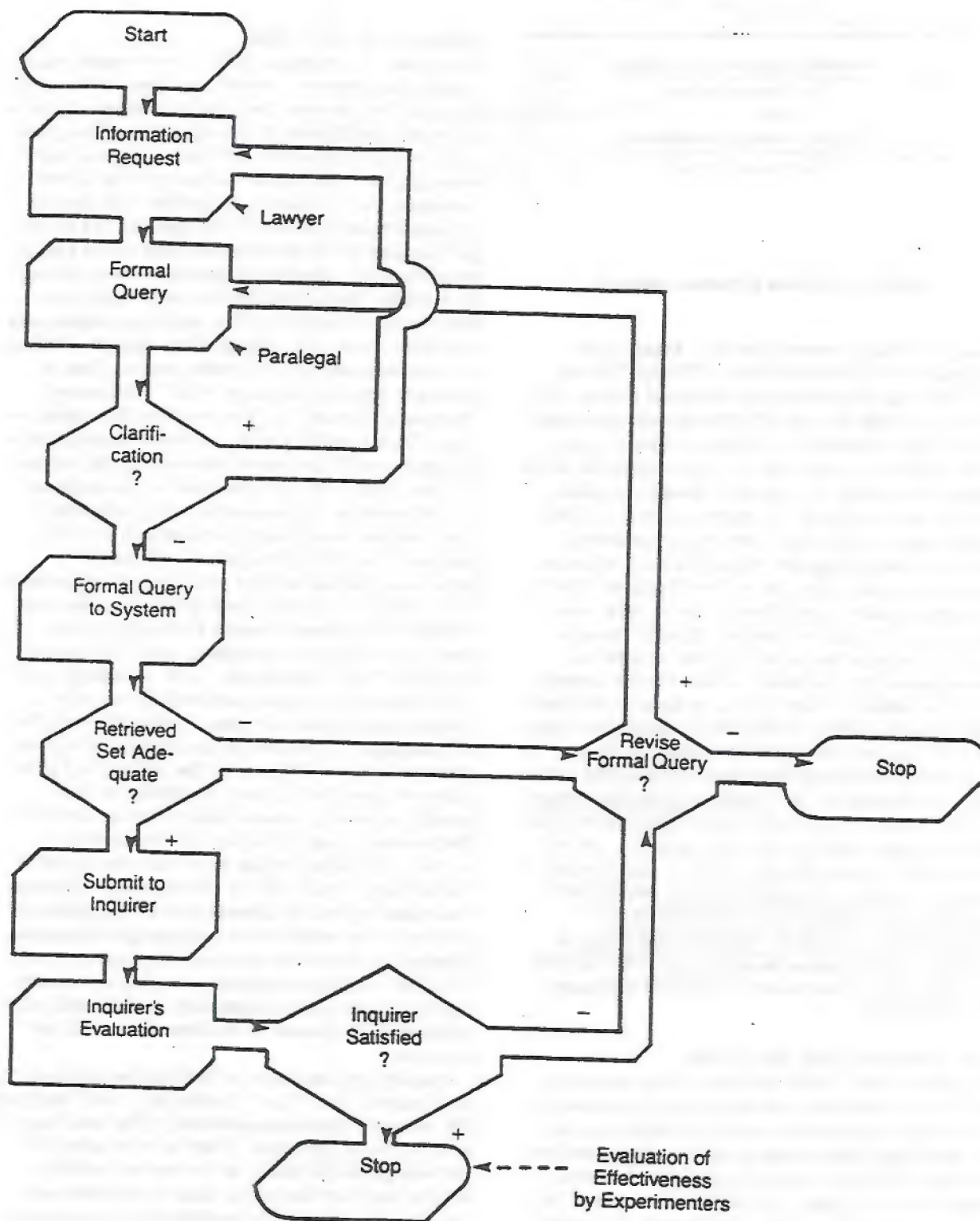


FIGURE 3. The Information Request Procedure

evant documents had been excluded). Random samples were taken from these subsets, and the samples were examined by the lawyers in a blind evaluation; the lawyers were not aware they were evaluating sample sets rather than retrieved sets they had personally gen-

erated. The total number of relevant documents that existed in these subsets could then be estimated. We sampled from subsets of the database rather than the entire database because, for most queries, the percentage of relevant documents in the database was less than

2 percent, making it almost impossible to have both manageable sample sizes and a high level of confidence in the resulting Recall estimates. Of course, no extrapolation to the entire database could be made from these Recall calculations. Nonetheless, the estimation of the number of relevant unretrieved documents in the subsets did give us a maximum value for Recall for each request.

TEST RESULTS

Of the 51 retrieval requests processed, values of Precision and Recall were calculated for 40. The other 11 requests were used to check our sampling techniques and control for possible bias in the evaluation of retrieved and sample sets.

In Table I we show the values of Precision and Recall for each of the 40 requests. The values of Precision ranged from a maximum of 100.0 percent to a minimum of 19.6 percent. The unweighted average value of Precision turned out to be 79.0 percent (standard deviation = 23.2). The weighted average was 75.5 percent. This meant that, on average, 79 out of every 100 documents retrieved using STAIRS were judged to be relevant.

The values of Recall ranged from a maximum of 78.7 percent to a minimum of 2.8 percent. The unweighted average value of Recall was 20 percent (standard deviation = 15.9), and the weighted average value was 20.26

percent. This meant that, on average, STAIRS could be used to retrieve only 20 percent of the relevant documents, whereas the lawyers using the system believed they were retrieving a much higher percentage (i.e., over 75 percent).

When we plot the value of Precision against the corresponding value of Recall for each of the 40 information requests, we get the scatter diagram given in Figure 4. Although Figure 4 contains no more data than Table I, it does show the relationships in a more explicit way. For example, the heavy clustering of points in the lower right corner shows that in over 50 percent of the cases we get values of Precision above 80 percent with Recall at or below 20 percent. The clustering in the lower portion of the diagram shows that in 80 percent of the information requests the value of Recall was at or below 20 percent. Figure 4 also depicts the frequently observed inverse relationship between Recall and Precision, where high values of Precision are often accompanied by low values for Recall, and vice versa [8].

OTHER FINDINGS

After the initial Recall/Precision estimations were done, several other statistical calculations were carried out in the hope that additional inferences could be made. First, the results were broken down by lawyer to ascertain whether certain individuals were *prima facie*

TABLE I. Recall and Precision Values for Each Information Request

Information request number	Recall	Precision	Information request number	Recall	Precision
1	.	.	27	50.0%	42.6%
2	45.5%	92.6%	28	50.0	19.6
3	.	.	29	.	.
4	.	.	30	7.0	100.0
5	.	.	31	.	.
6	8.9	60.0	32	12.5	100.0
7	20.6	64.7	33	18.2	79.5
8	43.9	88.8	34	14.1	45.1
9	13.3	48.9	35	.	.
10	10.4	96.8	36	4.2	33.3
11	12.8	100.0	37	15.9	81.8
12	9.6	84.2	38	24.7	68.3
13	15.1	85.0	39	18.5	83.3
14	78.7	99.0	40	4.1	100.0
15	.	.	41	18.3	96.9
16	.	.	42	45.4	91.0
17	.	.	43	18.9	100.0
18	13.0	38.0	44	10.6	100.0
19	15.8	42.1	45	20.3	94.0
20	19.4	68.9	46	11.0	85.7
21	41.0	33.8	47	13.4	100.0
22	22.2	94.8	48	13.7	87.5
23	2.8	100.0	49	17.4	87.8
24	.	.	50	13.5	75.7
25	13.0	94.0	51	4.7	100.0
26	7.2	95.0			

Average Recall = 20.0% (Standard deviation = 15.9)
Average Precision = 79.0% (Standard deviation = 23.3)

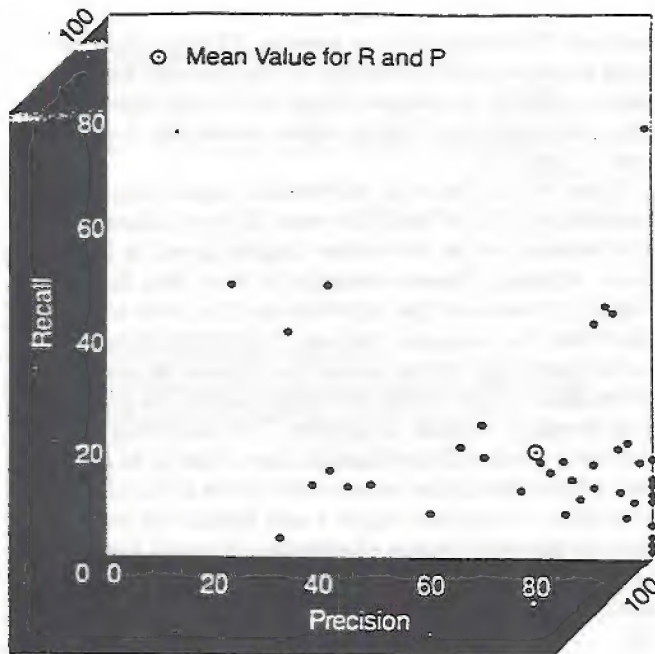


FIGURE 4. Plot of Precision versus Recall for All Information Requests

more adept at using the system than others. The results were as follows:

	Recall	Precision
Lawyer 1	22.7%	76.0%
Lawyer 2	18.0%	81.4%

Although there is some difference between the results for each lawyer, the variance is not statistically significant at the .05 level. Although this was a very limited test, we can conclude that at least for this experiment the results were independent of the particular user involved.

Another area of interest related to the revisions made to requests when the lawyer was not completely satisfied with the initial retrieved sets of documents. We hypothesized that if the values of Recall and Precision for the requests where substantial revisions had to be made (about 30 percent of the total) were significantly different from the overall mean values we might be able to infer something about the requesting procedure. Unfortunately, the values for Recall and Precision for the substantially revised queries (23.9 percent and 62.1 percent, respectively) did not indicate a statistically significant difference.

Finally, we tested the hypothesis that extremely high values of Precision for the retrieved sets would correlate directly with the lawyers' judgments of satisfaction with that set of documents (which might indicate that the lawyers were confusing Precision with Recall). To do this, we computed the mean Precision for all requests where the lawyers were satisfied with the initial retrieved set, and compared this value to the mean Precision for all requests. Although the Precision for requests that were not revised came out to be 85.4

percent, again the results were not statistically significant at the .05 level.

The Retrieval Effectiveness of Lawyers versus Paralegals

The argument can be made that, because STAIRS is a high-speed, on-line, interactive system, the searcher at the terminal can quickly and effectively evaluate the output of STAIRS during the query modification process. Therefore, retrieval effectiveness might be significantly improved if the person originating the information request is actually doing the searching at the terminal. This would mean that if a lawyer worked directly on the query formulation and query modification at the STAIRS terminal, rather than using a paralegal as intermediary, retrieval effectiveness might be improved.

We tested this conjecture by comparing the retrieval effectiveness of the lawyer vis à vis the paralegal on the same information request. We selected (at random) five information requests for which the searches had already been completed by the paralegal, and for which retrieved sets had been evaluated by the lawyer and values of Recall computed. (Neither the lawyer who made the relevance judgments nor the paralegal knew the Recall figures for these original requests.) We invited the lawyer to use STAIRS directly to access the database, giving the lawyer copies of his or her original information requests. The lawyer translated these requests into formal queries, evaluating the text displayed on the screen, modifying the queries as he or she saw fit, and finally deciding when to terminate the search. For each of the five information requests, we estimated the minimum number of relevant documents in the entire file, and knowing which documents the lawyer had previously judged relevant, we were able to compute the values of Recall for the lawyer at the terminal as we had already done for the paralegal. If it were true that STAIRS would give better results when the lawyers themselves worked at the terminal, the values of Recall for the lawyers would have to be significantly higher than the values of Recall when the paralegals did the searching. The results were as follows:

Request number	Recall (paralegal)	Recall (lawyer)
1	7.2%	6.6%
2	19.4%	10.3%
3	4.2%	26.4%
4	4.1%	7.4%
5	18.9%	25.3%
Mean	10.7%	15.2%
	(s.d. = 7.65)	(s.d. = 9.83)

Although there is a marked improvement in the lawyer's Recall for requests 3, 4, and 5, and in the average Recall for all five information requests, the improvement is not statistically significant at the .05 level

($z = -0.81$). Hence, we cannot reject the hypothesis that

($z = -0.81$)

both the lawyer and the paralegal get the same results for Recall.

WHY WAS RECALL SO LOW

The realization that STAIRS may be retrieving only one out of five relevant documents in response to an information request may surprise those who have used STAIRS or had it demonstrated to them. This is because they will have seen only the retrieved set of documents and not the total corpus of relevant documents; that is, they have seen that the proportion of relevant documents in the retrieved set (i.e., Precision) is quite good (around 80 percent). The important issues to consider here are (1) why was Recall so low and (2) why did the users (lawyers and paralegals) believe they were retrieving 75 percent of the relevant documents when, in fact, they were only retrieving 20 percent.

The low values of Recall occurred because full-text retrieval is difficult to use to retrieve documents by subject because its design is based on the assumption that it is a simple matter for users to foresee the exact words and phrases that will be used in the documents they will find useful, and *only* in those documents. This assumption is not a new one; it goes back over 25 years to the early days of computing. The basic idea is that one can use the formal aspects of text to predict its meaning or subject content: formal aspects such as the occurrence, location, and frequency of words; and to the extent that it can be precisely described, the syntactic structure of word phrases. It was hoped that by exploiting the high speed of a computer to analyze the formal aspects of text, one could get the computer to deal with text in a "comprehending-like" way (i.e., to identify the subject content of texts). This endeavor is known as "Automatic Indexing" or, in a more general sense, "Natural Language Processing." During the past two decades, many experiments in automatic indexing (of which full-text searching is the simplest form) have been carried out, and many discussions by linguists, psychologists, philosophers, and computer scientists have analyzed the results and the issues [5]. These experiments show that full-text document retrieval has worked well only on unrealistically small databases.

The belief in the predictability of the words and phrases that may be used to discuss a particular subject is a difficult prejudice to overcome. In a naive sort of way, it is an appealing prejudice but a prejudice nonetheless, because the effectiveness of full-text retrieval has not been substantiated by reliable Recall measures on realistically large databases. Stated succinctly, it is impossible for users to predict the exact words, word combinations, and phrases that are used by *all* (or most) relevant documents and *only* (or primarily) by those documents, as can be seen in the following examples. See FURBAS et al. 1983 *Semantic Retrieval*.

In the legal case in question, one concern of the lawyers was an accident that had occurred and was now an object of litigation. The lawyers wanted all the reports, correspondence, memoranda, and minutes of meetings that discussed this accident. Formal queries

were constructed that contained the word "accident(s)" along with several relevant proper nouns. In our search for *unretrieved* relevant documents, we later found that the accident was not always referred to as an "accident," but as an "event," "incident," "situation," "problem," or "difficulty," often without mentioning any of the relevant proper names. The manner in which an individual referred to the incident was frequently dependent on his or her point of view. Those who discussed the event in a critical or accusatory way referred to it quite directly—as an "accident." Those who were personally involved in the event, and perhaps culpable, tended to refer to it euphemistically as, *inter alia*, an "unfortunate situation," or a "difficulty." Sometimes the accident was referred to obliquely as "the subject of your last letter," "what happened last week was . . ." or, as in the opening lines of the minutes of a meeting on the issue, "Mr. A: We all know why we're here . . ." Sometimes relevant documents dealt with the problem by mentioning only the technical aspects of why the accident occurred, but neither the accident itself nor the people involved. Finally, much relevant information discussed the situation *prior* to the accident and, naturally, contained no reference to the accident itself.

Another information request resulted in the identification of 3 key terms or phrases that were used to retrieve relevant information; later, we were able to find 26 other words and phrases that retrieved additional relevant documents. The 3 original key terms could not have been used individually as they would have retrieved 420 documents, or approximately 4000 pages of hard copy, an unreasonably large set, most of which contained irrelevant information. Another request identified 4 key terms/phrases that retrieved relevant documents, which we were later able to enlarge by 44 additional terms and combinations of terms to retrieve relevant documents that had been missed.

Sometimes we followed a trail of linguistic creativity through the database. In searching for documents discussing "trap correction" (one of the key phrases), we discovered that relevant, unretrieved documents had discussed the same issue but referred to it as the "wire warp." Continuing our search, we found that in still other documents trap correction was referred to in a third and novel way: the "shunt correction system." Finally, we discovered the inventor of this system was a man named "Coxwell" which directed us to some documents he had authored, only he referred to the system as the "Roman circle method." Using the Roman circle method in a query directed us to still more relevant but unretrieved documents, but this was not the end either. Further searching revealed that the system had been tested in another city, and all documents germane to those tests referred to the system as the "air truck." At this point the search ended, having consumed over an entire 40-hour week of on-line searching, but there is no reason to believe that we had reached the end of the trail; we simply ran out of time.

As the database included many items of personal cor-

respondence as well as the verbatim minutes of meetings, the use of slang frequently changed the way in which one would "normally" talk about a subject. Disabled or malfunctioning mechanisms with which the lawsuit was concerned were sometimes referred to as "sick" or "dead," and a burned-out circuit was referred to as being "fried." A critical issue was sometimes referred to as the "smoking gun."

Even misspellings proved an obstacle. Key search terms like "flattening," "gauge," "memos," and "correspondence," which were essential parts of phrases, were used effectively to retrieve relevant documents. However, the misspellings "flatening," "guage," "gage," "memoes," and "correspondance," using the same phrases, also retrieved relevant documents. Misspellings like these, which are tolerable in normal everyday correspondence, when included in a computerized database become literal traps for users who are asked not only to anticipate the key words and phrases that may be used to discuss an issue but also to foresee the whole range of possible misspellings, letter transpositions, and typographical errors that are likely to be committed.

Some information requests placed almost impossible demands on the ingenuity of the individual constructing the query. In one situation, the lawyer wanted "Company A's comments concerning . . ." Looking at the documents authored by Company A was not enough, as many relevant comments were embedded in the minutes of meetings or recorded secondhand in the documents authored by others. Retrieving all the documents in which Company A was mentioned was too broad a search; it retrieved over 5,000 documents (about 40,000+ pages of hard copy). However, predicting the exact phraseology of the text in which Company A commented on the issue was almost impossible: sometimes Company A was not even mentioned, only that so-and-so (representing Company A) "said/considered/remarked/pointed out/commented/noted/explained/discussed," etc.

In some requests, the most important terms and phrases were not used at all in relevant documents. For example, "steel quantity" was a key phrase used to retrieve important relevant documents germane to an actionable issue, but unretrieved relevant documents were also found that did not report *steel quantity* at all, but merely the *number* of such things as "girders," "beams," "frames," "bracings," etc. In another request, it was important to find documents that discussed "non-expendable components." In this case, relevant unretrieved documents merely listed the names of the components (of which there were hundreds) and made no mention of the broader generic description of these items as "nonexpendable."

Why didn't the lawyers realize they were not getting all of the information relevant to a particular issue? Certainly they knew the lawsuit. They had been involved with it from the beginning and were the principal attorneys representing the defense. In addition, one of the paralegals had been instrumental not only in setting up the database but also in supervising the se-

lection of relevant information to be put on-line. Might it not be reasonable to expect them to be suspicious that they were not retrieving everything they wanted? Not really. Because the database was so large (providing access to over 350,000 pages of hard copy, all of which was in some way pertinent to the lawsuit), it would be unreasonable to expect four individuals (two lawyers and two paralegals) to have total recall of all the important supporting facts, testimony, and related data that were germane to the case. If they had such recall they would have no need for a computerized, interactive retrieval system. It is well known among cognitive psychologists that man's power of literal recall is much less effective than his power of recognition. The lawyers could remember the exact text of some of the important information, but as we have already stated, this was a very small subset of the total information relevant to a particular issue. They could *recognize* the important information when they saw it, and they could do so with uncanny consistency. (As a control, we submitted some retrieved sets and sample sets of documents to the lawyers several times in a blind test of their evaluation consistency, and found that their consistency was almost perfect.) Also, since the lawyers were not experts in information retrieval system design, there were no a priori reasons for them to suspect the Recall levels of STAIRS.

DETERIORATION OF RECALL AS A FUNCTION OF FILE SIZE

One reason why Recall evaluations done on small databases cannot be used to estimate Recall on larger databases is because, *ceteris paribus*, the value of Recall decreases as the size of the database increases, or, from a different point of view, the amount of search effort required to obtain the same Recall level increases as the database increases, often at a faster rate than the increase in database size. On the database we studied, there were many search terms that, used by themselves, would retrieve over 10,000 documents. Such output overload is a frequent problem of full-text retrieval systems.

As a retrieved set of several thousand documents is impractical, the user must reduce the output overload by reformulating the single-term query so that it retrieves fewer documents. If a single term query w_1 retrieves too many documents, the user may add another term, w_2 , so as to form the new query " w_1 and w_2 " (or " w_1 adjacent w_2 ," or " w_1 same w_2 "). The reformulated query cannot retrieve more documents than the original; most probably, it will retrieve many fewer. The process of adding intersecting terms to a query can be continued until the size of the output reaches a manageable number. (This strategy, and its consequences, is discussed in more detail in [1].) However, as the user narrows the size of the output by adding intersecting terms, the value of Recall goes down because, with each new term, the probability is that some relevant documents will be excluded by that reformulated query.

The deterioration of Recall from a probabilistic point of view is quite startling. For each query, there is a class of relevant documents that we designate as R . We represent the probability that each of those documents will contain some word w_1 as p , and the probability that a relevant document will contain some other word w_2 as q . Thus, the value of Recall for a request using only w_1 will be equal to p , and Recall for a request using only w_2 will be equal to q . Now the probability that a relevant document will contain both w_1 and w_2 is less than or equal to either p or q . If we assume that the respective appearances of w_1 and w_2 in a relevant document are independent events, then the probability of both of them appearing in a relevant document would be equal to the product of p and q . Since both p and q are usually numbers less than unity, their product usually will be smaller than either p or q . This means that Recall, which can also be thought of as the probability of retrieving a relevant document, is now equal to the product of p and q . In other words, reducing the number of documents retrieved by intersecting an increasing number of terms in the formal query causes Recall for that query also to decrease.

However, the problem is really much worse. In order for a relevant document, which contains w_1 and w_2 , to be retrieved by a single query, a searcher must select and use those words in his or her query. The probability that the searcher will select w_1 is, of course, generally less than 1.0; and the probability that w_1 will occur in a relevant document is also usually less than 1.0. However, these probabilities must be multiplied by the probability that the searcher will select w_2 as part of his or her query, and the probability that w_2 will occur in a relevant document. Thus, calculating Recall for a two-term search involves the multiplication of four numbers each of which is usually less than 1.0. As a result, the value of Recall gets very small (see Table II). When

TABLE II. The Probability of Retrieving a Relevant Document Containing Terms w_1 and w_2

$P(Sw_1) = .6$ = Probability searcher uses term w_1 in a search query
$P(Sw_2) = .5$ = Probability searcher uses term w_2 in a search query
$P(Dw_1) = .7$ = Probability w_1 appears in a relevant document
$P(Dw_2) = .6$ = Probability w_2 appears in a relevant document
Probability of searcher selecting w_1 and a relevant document containing w_1 :
$P(Sw_1) \times P(Dw_1) = (.6) \times (.7) = .42$
Probability of searcher selecting w_2 and a relevant document containing w_2 :
$P(Sw_2) \times P(Dw_2) = (.5) \times (.6) = .30$
Probability of searcher selecting w_1 and w_2 and a relevant document containing w_1 and w_2 :
$P(Sw_1) \times P(Dw_1) \times P(Sw_2) \times P(Dw_2)$
(e.g., $P(.6) \times P(.7) \times P(.5) \times P(.6) = .126$)

The sad truth of IR!

we consider a three- or four-term query, the value of Recall drops off even more sharply.

The problem of output overload is especially critical in full-text retrieval systems like STAIRS, where the frequency of occurrence of search terms is considerably larger than (and increases faster than) the frequency of occurrence (or "breadth") of index terms in a database where the terms are manually assigned to documents. This means that the user of a full-text retrieval system will face the problem of output overload sooner than the user of a manually indexed system. The solution that STAIRS offers—conjunctively adding search terms to the query—does reduce the number of documents retrieved to a manageable number but also eliminates relevant documents. Search queries employing four or five intersecting terms were not uncommon among the queries used in our test. However, the probability that a query that intersects five terms will retrieve relevant documents is quite small. If we were to assign a probability of .7 to all the respective probabilities in a hypothetical five-term query as we did in the two-term query in Table II (and .7 is an optimistic average value), the Recall level for that query would be .028. In other words, that query could be expected to retrieve less than 3 percent of the relevant documents in the database. If the probabilities for the five-term query were a more realistic average of .5, the Recall value for that query would be .0009! This means that if there were 1000 relevant documents on the database, it is likely that this query would retrieve only one of them. The searcher must submit many such low-yield queries to the system if he or she wants to retrieve a high percentage of the relevant documents.

DISCUSSION

The reader who is surprised at the results of this test of retrieval effectiveness is not alone. The lawyers who participated in the test were equally astonished. Although there are sound theoretical reasons why we should expect these results, they seem to run counter to previous tests of retrieval effectiveness for full-text retrieval.

Two pioneering evaluations of full-text retrieval systems by respected researchers in the field (Swanson [6] and Salton [3]) determined to their satisfaction that full-text document-retrieval systems could retrieve relevant documents at a satisfactory level while avoiding the problems of manual indexing. Our study, on the other hand, shows that full-text document retrieval does *not* operate at satisfactory levels and that there are sound theoretical reasons to expect this to be so. Who is right? Well, we all are, and this is not an equivocation. The two earlier studies drew the correct conclusions from their evaluations, but these conclusions were different from ours because they were based on small experimental databases of less than 750 documents. Our study was done not on an experimental database but an actual, operational database of almost 40,000 documents. Had Swanson and Salton been fortunate enough to study a retrieval system as large as ours, they

would undoubtedly have observed similar phenomena (Swanson was later to comment perceptively on the difficulty of drawing accurate conclusions about document retrieval from experiments using small databases [7]). In addition, it has only recently been observed that information-retrieval systems do not scale up [2]. That is, retrieval strategies that work well on small systems do not necessarily work well on larger systems (primarily because of output overload). This means that studies of retrieval effectiveness must be done on full-sized retrieval systems if the results are to be indicative of how a large, operational system would perform. However, large-scale, detailed retrieval-effectiveness studies, like the one reported here, are unprecedented because they are incredibly expensive and time consuming; our experiment took six months; involved two researchers and six support staff; and, taking into account all direct and indirect expenses, cost almost half a million dollars. Nevertheless, Swanson and Salton's earlier full-text evaluations remain pioneering studies and, rather than contradict our findings, have an illuminating value of their own.

An objection that might be made to our evaluation of STAIRS is that the low Recall observed was not due to STAIRS but rather to query-formulation error. This objection is based on the realization that, at least in principle, virtually any subset of the database is retrievable by some simple or complex combination of search terms. The user's task is simply to find the right combination of search terms to retrieve *all* and *only* the relevant documents. However, we believe that users should not be asked to shoulder the blame, and perhaps an analogy will indicate why. Suppose you ask a company to make a lock for you, and they oblige by providing a combination lock; but when you ask them for the combination to open the lock, they say that finding the correct combination is your problem, not theirs. Now, it is possible, in principle, to find the correct combination, but in practice it may be impossibly difficult to do so. A full-text retrieval system bears the burden of retrieval failure because it places the user in the position of having to find (in a relatively short time) an impossibly difficult combination of search terms. The person using a full-text retrieval system to find information on a relatively large database is in the same unenviable position as the individual looking for the combination to the lock. It is true that we, as evaluators, found the combinations of search terms necessary to retrieve many of the unretrieved relevant documents, but three things should be kept in mind. First, we make no claim to having found all the relevant unretrieved documents; we may not have found even half of them, as our sampling technique covered only a small percentage of the database. Second, a tremendous amount of search time was involved with each request (sometimes over 40 hours of on-line time), and the entire test took almost 6 months. Such inefficiency is clearly not consonant with the high speed desired for computerized retrieval. Third, the evaluators in this case represented, together, over 40 years of practical and theoretical ex-

perience in information systems analysis and should be expected to have somewhat better searching abilities than the typical STAIRS searcher. Moreover, STAIRS is sold under the premise that it is easy to use and requires no sophisticated training on the part of the user. Yet this study is a clear demonstration of just how sophisticated search skills must be to use STAIRS, or, mutatis mutandis, any other full-text retrieval system. There is evidence that this problem is beginning to be recognized by at least one full-text retrieval vendor, WESTLAW, which has made its reputation by offering full-text access to legal cases. WESTLAW has now begun to supplement its full-text retrieval with manually assigned index terms.

SUMMARY

This paper has presented a major, detailed evaluation of a full-text document-retrieval system. We have shown that the system did not work well in the environment in which it was tested and that there are theoretical reasons why full-text retrieval systems applied to large databases are unlikely to perform well in any retrieval environment. The optimism of early studies was based on the small size of the databases used, and were geared toward showing only that full-text search was *competitive* with searching based on manually assigned index terms, under the assumption that, if it were competitive, full-text retrieval would eliminate the cost of indexing. However, there are costs associated with a full-text system that a manual system does not incur. First, there is the increased time and cost of entering the full text of a document rather than a set of manually assigned subject and context descriptors. The average length of a document record on the system we evaluated was about 10,000 characters. In a manually assigned index-term system of the same type, we found the average document record to be less than 500 characters. Thus, the full-text system incurs the additional cost of inputting and verifying 20 times the amount of information that a manually indexed system would need to deal with. This difference alone would more than compensate for the added time needed for manual indexing and vocabulary construction. The 20-fold increase in document record size also means that the database for a full-text system will be some 20 times larger than a manually indexed database and entail increased storage and searching costs. Finally, because the average number of searchable subject terms per document for the full-text retrieval system described here was approximately 500, whereas a manually indexed system might have a subject indexing depth of about 10, the dictionary that lists and keeps track of these assignments (i.e., provides pointers to the database) could be as much as 50 times larger on a full-text system than on a manually indexed system. A full-text retrieval system does not give us something for nothing. Full-text searching is one of those things, as Samuel Johnson put it so succinctly, that "... is never done well, and one is surprised to see it done at all."

Acknowledgments. The authors would like to thank William Cooper of the University of California at Berkeley for his comments on an earlier version of this manuscript, and Barbara Blair for making the drawings that accompany the text.

REFERENCES

1. Blair, D.C. Searching biases in large interactive document retrieval systems. *J. Am. Soc. Inf. Sci.* 31 (July 1980), 271-277.
2. Resnikoff, H.L. The national need for research in information science. STI Issues and Options Workshop, House subcommittee on science, research and technology, Washington, D.C., Nov. 3, 1978.
3. Salton, G. Automatic text analysis. *Science* 168, 3929 (Apr. 1970), 335-343.
4. Saracevic, T. Relevance: A review of and a framework for thinking on the notion in information science. *J. Am. Soc. Inf. Sci.* 26 (1975), 321-343.
5. Sparck Jones, K. *Automatic Keyword Classification for Information Retrieval*. Butterworths, London, 1971.
6. Swanson, D.G. Searching natural language text by computer. *Science* 132, 3434 (Oct. 1960), 1099-1104.
7. Swanson, D.R. Information retrieval as a trial and error process. *Libr. Q.* 47, 2 (1978), 128-148.
8. Swets, J.A. Information retrieval systems. *Science* 141 (1963), 245-250.
9. Zunde, P., and Dexter, M.E. Indexing consistency and quality. *Am. Doc.* 20, 3 (July 1969), 259-264.

CR Categories and Subject Descriptors: H.1.0 [Models and Principles]: General; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—search process, query formulation
General Terms: Design, Human Factors, Theory
Additional Key Words and Phrases: full-text document retrieval, litigation support, retrieval evaluation, Recall and Precision

Received 4/84; accepted 9/84

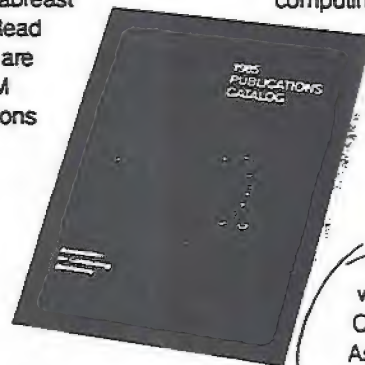
Authors' Present Addresses: David C. Blair, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI 48109; M.E. Maron, School of Library and Information Studies, The University of California, Berkeley, CA 94720.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

SUBSCRIBE TO ACM PUBLICATIONS

Whether you are a computing novice or a master of your craft, ACM has a publication that can meet your individual needs. Do you want broad-gauge, high quality, highly readable articles on key issues and major developments and trends in computer science? Read *Communications of the ACM*. Do you want to read comprehensive surveys, tutorials, and overview articles on topics of current and emerging importance? *Computing Surveys* is right for you. Are you interested in a publication that offers a range of scientific research designed to keep you abreast of the latest issues and developments? Read *Journal of the ACM*. What specific topics are worth exploring further? The various ACM transactions cover research and applications

in-depth—ACM Transactions on Mathematical Software, ACM Transactions on Database Systems, ACM Transactions on Programming Languages and Systems, ACM Transactions on Graphics, ACM Transactions on Office Information Systems, and ACM Transactions on Computer Systems. Do you need additional references on computing? *Computing Reviews* contains original reviews and abstracts of current books and journals. The *ACM Guide to Computing Literature* is an important bibliographic guide to computing literature. *Collected Algorithms from ACM* is a collection of ACM algorithms available in printed version, on microfiche, or machine-readable tape.



For more information about ACM publications, write for your free copy of the ACM Publications Catalog to: The Publications Department, The Association for Computing Machinery, 11 West 42nd Street, New York, NY 10036.

Neural Networks: Applications in Industry, Business and Science

BERNARD WIDROW ■ DAVID E. RUMELHART ■ MICHAEL A. LEHR

Just four years ago, the only widely reported commercial application of neural network technology outside the financial industry was the airport baggage explosive detection system [27] developed at Science Applications International Corporation (SAIC). Since that time scores of industrial and commercial applications have come into use, but the details of most of these systems are considered corporate secrets and are shrouded in secrecy. This hastening trend is due in part to the availability of an increasingly wide array of dedicated neural network hardware. This hardware is either in the form of accelerator cards for PCs and workstations or a large number of integrated circuits implementing digital and analog neural networks either currently available or in the final stages of design. An assortment of tools and development systems is provided by the manufacturers of most of these products.

Complementing the hardware are scores of commercial software packages now available. Many packages can be quickly tailored to provide low-cost turnkey solutions to a broad spectrum of applications. A very useful list containing 64 of these software and hardware tools together with their prices and the names, addresses, and phone numbers of the vendors is published in a recent issue of the magazine *PC AI* [17]. Other valuable lists of neural network tools and vendors can be found in the February issue of *Dr. Dobb's Journal* [11] and the June 1992 issue of *AI Expert*. That these lists are not complete is an indication of the rapid growth the field is presently enjoying. It is not possible in a short article to cite all of

the existing applications. The examples described herein are meant only to be representative samples.

Linear Neural Network Applications

The first successful applications of adaptive neural networks were developed by Widrow and Hoff in the 1960s. They employed single-neuron linear networks trained by the LMS algorithm [32]. Single-element and multielement linear networks are equally easy to train and have found widespread commercial application over the past three decades. A few of these applications include:

- **Telecommunications.** Modems used in the high-speed transmission of digital data through telephone

channels use adaptive line equalizers and adaptive echo cancellers. Each adaptive system utilizes a single-neuron neural network. The most significant commercial application of neural networks today is in this area.

- **Control of sound and vibration.** Active control of vibration and noise is accomplished by using an adaptive actuator to generate equal and opposite vibration and noise. This is being used in air-conditioning systems, in automotive systems, and in industrial applications.

- **Particle accelerator beam control.** The Stanford Linear Accelerator Center (SLAC) is now using adaptive techniques to cancel disturbances that diminish the positioning accuracy of opposing beams of positrons

*Gerber Baby Foods uses **neural networks** to
 help manage its trade in **cattle futures**... Spiegel is using
 software to determine which customers get their **catalogs**.*



and electrons in a particle collider. The accuracy is being held to within 2 microns in order to have a satisfactory number of collisions. The efficiency of this 3-kilometer long, billion dollar machine is being enhanced by the use of linear adaptive noise cancelling.

Multielement Nonlinear Network Applications

Unlike their linear counterparts which have a long track record of success, nonlinear multielement neural networks have begun proving themselves in commercial applications only recently. This is largely because the most useful neural network algorithm—backpropagation—did not become widely known until 1986, when it was published in Rumelhart and McClelland's two-volume PDP set [21]. Also important in the timing of the current boom in nonlinear neural network applications has been the rapid advance of computer and microprocessor performance, which continues to improve the feasibility and cost-effectiveness of computationally intensive algorithms. Although nonlinear neural networks are not currently being used as widely as linear networks, they are applicable to a much broader range of problems than their linear counterparts. Furthermore, the applications for which they are best suited often involve complex nonlinear relationships for which acceptable classical solutions are unavailable.

Successful commercial applications of nonlinear multielement neural networks in most cases currently rely on the backpropagation algorithm, with some use of backpropagation-through-time [30], radial basis functions [11], genetic algorithms [3, 24], Kohonen's Learning Vector Quantization (LVQ) [9], and a number of other algorithms. Whatever the paradigm, neural networks are currently being used

throughout business and industry to satisfy a diverse assortment of needs. Most neural network applications address problems described by one of the following three categories: 1) pattern classification, 2) prediction and financial analysis, and 3) control and optimization. Examples from each category follow:

Pattern Classification

Credit card fraud detection. Several banks and credit card companies including American Express, Mellon Bank, First USA Bank, and others are currently using neural networks to study patterns of credit card usage and to detect transactions that are potentially fraudulent [8, 10, 26]. Credit card fraud is a growing problem that threatens the entire industry. Some institutions are using home-grown software, while others are using commercial products developed by Nestor, HNC, and other companies.

Machine-printed character recognition. Commercial products performing machine-printed character recognition have been introduced by a large number of companies and have been described in the literature. Among these products are those made by Sharp Corp. [9, 26], Mitsubishi Electric Corp. [9], VeriFone Inc. [8, 9, 11, 26], Hecht-Nielsen Corp. (HNC) [11], Nestor Inc. [33], Calera Recognition Systems Inc. [11], Caere Corp. [11], and Audre Recognition Systems [11]. Sharp's Optical Character Recognition (OCR) system is used to recognize Japanese characters. It contains approximately 10 million weights and uses a variant of Kohonen's LVQ algorithm. It outperforms existing conventional systems in speed and accuracy. Mitsubishi is currently developing a similar system [9]. VeriFone's Onyx Check Reader provides an accurate, low-cost system for reading identification numbers on checks by using a custom analog neural net chip made by Syn-

aptics. Calera Recognition Systems markets a product, FaxGrabber, which automatically converts incoming faxes to text using a modified radial basis function neural network to perform OCR. Highlighting the secrecy with which many firms guard their reliance on neural network technology, Calera did not acknowledge their use of the technology (which began in 1986) until 1992 when competitor Caere Corp. announced the use of neural nets in Caere's highly successful AnyFax OCR engine. AnyFax is used in Caere's FaxMaster software and is licensed for use in other products including Delrina Technology Inc.'s WinFax Pro 3.0 fax software. Audre Recognition Systems uses a variant of the backpropagation algorithm in its OCR product, the Audre Neural Network, which not only reads standard alphanumeric characters but can also be trained to recognize specialized symbols on engineering drawings [11].

Hand-printed character recognition. HNC's Quickstrokes Automated Data Entry System is being used to recognize handwritten forms at Avon's order-processing center and at the state of Wyoming's Department of Revenue. In the June 1992 issue of *Systems Integration Business*, Dennis Livingston reports that before implementing the system, Wyoming was losing an estimated \$300,000 per year in interest income because so many checks were being deposited late. Cardiff Software offers a product called Teleform which uses Nestor's hand-printed character recognition system to convert a fax machine into an OCR scanner. Poquet Computer, now a subsidiary of Fujitsu, uses Nestor's NestorWriter neural network software to perform handwriting recognition for the pen-based PC it announced in January 1992 [25].

Cursive handwriting recognition. Neural networks have proved useful in the development of algorithms for

on-line cursive handwriting recognition [20]. A recent startup company in Palo Alto, Lexicus, beginning with this basic technology has developed an impressive PC-based cursive handwriting system.

Quality control in manufacturing. Neural networks are being used in a large number of quality control and quality assurance programs throughout industry. Applications include contaminant-level detection from spectroscopy data at chemical plants [11, 14] and loudspeaker defect classification by CTS Electronics [1]. According to Justin Kestelyn in the June 1990 issue of *AI Expert*, neural networks are also being used by the Florida Department of Citrus to perform orange juice purity evaluation. Applied Intelligent Systems of Ann Arbor, Mich., has built into its vision computers neural recognition features that are used for quality control in factories [11].

Event detection in particle accelerators. Research into the feasibility of using neural networks to detect notable events in high-energy particle colliders has been performed at the European Center for Particle Physics (CERN), and at a number of other research organizations [5]. Steven Kasow of CERN has reported that scientists there are using fast analog neural networks in real-time triggering systems for detectors. This permits the distillation of an enormous number of candidate events into a manageable set of "interesting" events which can be recorded on mass-storage devices and studied further. Neural networks are proving especially useful and cost-effective when used in experiments for which complex criteria are needed to differentiate between interesting and uninteresting events. Similar work is taking place at the Fermi National Accelerator Laboratory, Batavia, Ill., using Intel's high-speed analog ETANN neural network chip, according to the June 1993 issue of the *Cognizer Report* newsletter.

Petroleum exploration. Oil companies including Arco and Texaco are using neural networks to help determine the locations of underground oil and gas deposits [25].

War on drugs. Yes, neural networks have even made their debut in

the U.S. government's famous war on drugs. PC-based software emulating a multilayer neural network is being used on a daily basis at the North Carolina State Bureau of Investigation (NCSBI) to help forensic experts identify cocaine samples originating from the same batch. J. F. Casale and J. W. Watterson report in the March 1993 issue of the *Journal of Forensic Sciences* that the information helps undercover agents put together drug-related criminal cases.

Medical applications. Commercial products by Neuromedical Systems, Inc. are used for cancer screening and other medical applications [8, 9, 11, 19, 26]. The company markets electrocardiograph and pap smear systems that rely on neural network technology. The pap smear system, *Papnet*, is able to help cytotechnologists spot cancerous cells, drastically reducing false/negative classifications. The system is used by the U.S. Food and Drug Administration [6].

Prediction and Financial Analysis

Financial forecasting and portfolio management. Neural networks are used for financial forecasting at a large number of investment firms and financial entities including Merrill Lynch & Co., Salomon Brothers, Shearson Lehman Brothers Inc., Citibank, and the World Bank [3, 9, 24, 25]. Gerber Baby Foods reportedly uses neural networks to help manage its trade in cattle futures [6]. Using neural networks trained by genetic algorithms, Citibank's Andrew Colin claims to be able to earn 25% returns per year investing in the currency markets. A startup company, Promised Land Technologies, offers a \$249 software package that is claimed to yield impressive annual returns [24].

Loan approval. Chase Manhattan Bank reportedly uses a hybrid system utilizing pattern analysis and neural networks to evaluate corporate loan risk. Robert Marose reports in the May 1990 issue of *AI Expert* that the system, Creditview, helps loan officers estimate the credit worthiness of corporate loan candidates.

Real estate analysis. HNC's Areas Automated Property Valuation System [8] is being used by Foster Ous-

ley Conley to evaluate the value of residential property in California.

Marketing analysis. The Target Marketing System developed by Churchill Systems is currently in use by Veratex Corp. to optimize marketing strategy and cut marketing costs by removing unlikely future customers from a list of potential customers [8]. Likewise, Spiegel Inc. is using software created by NeuralWare Inc. to determine which customers should receive their mail order catalogs. Spiegel's director of market research expects savings of at least \$1 million per year based on increased sales and reduced catalog mailings [25].

Airline seating allocation. The Airline Marketing Assistant/Tactician developed by BehavHeuristics Inc. uses neural networks to predict passenger demand and allocate seating for carriers including Nationair Canada and USAir [8].

Control and Optimization

Electric arc furnace electrode position control. Electric arc furnaces are used to melt scrap steel. The Intelligent Arc Furnace controller systems installed by Neural Applications Corp. [8, 28] are reportedly saving millions of dollars per year per furnace in increased furnace throughput and reduced electrode wear and electricity consumption. The controller is currently being installed at furnaces worldwide.

Semiconductor process control. Kopin Corp. has used neural networks to cut dopant concentration and deposition thickness errors in solar cell manufacturing by more than a factor of two [9].

Chemical process control. Pavilion Technologies has developed a neural network process control package, Process Insights, which is helping Eastman Kodak and a number of other companies reduce waste, improve product quality, and increase plant throughput [4, 8, 9, 11, 12]. Neural network models are being used to perform sensitivity studies, determine process set points, detect faults, and predict process performance.

Petroleum refinery process control. Texaco's Puget Sound Refinery, which processes 120,000 barrels of

oil a day, has integrated neural networks into the plant's process control systems. As described in the June 1990 issue of *AI Expert*, one of these networks is used in the control of a debutanizer, a system which separates hydrocarbons according to their molecular weights. This requires precise monitoring of temperatures, pressures, and flow rates. The 17-hour batch cycle subjects the process to constant instability. A neural network has been built and trained to help ensure product quality during periods of change and instability. The seven-input, two-output network, which was trained with roughly 1,500 data samples, is usually able to correct errors in the control parameters before they appear. A feedback mechanism helps reduce unexpected errors that do occur.

Continuous-casting control during steel production. A neural control system is in operation in Japan at plants owned by Fujitsu Ltd. and Nippon Steel Corp. The system has reduced costs by several million dollars a year by eliminating the damage and downtime caused by "breakout," when imperfect control allows spillage of molten steel [9, 26, 30]. The system uses a feedforward network trained by backpropagation to detect breakout before it occurs, allowing corrective measures to be taken. The control system has been operating since early 1990.

Food and chemical formulation optimization. Neural networks are used to optimize formulations at the Glidden Co., the Lord Corp. [7], and at M&M/Mars. Researchers at the first two companies report success using AI Ware's CAD/Chem package to search for improved chemical formulations. CAD/Chem has been used by Lord Corp. in the process of formulating a new adhesive product [7] by an iterative search technique.

Nonlinear Applications on the Horizon

A large number of research programs are developing neural network solutions that are either likely to be used in products in the near future or, particularly in the case of military applications, that may already be incorporated into products,

albeit unadvertised. This category is much larger than the foregoing, so we present here only a few representative examples

Missile guidance and detonation. David Andes at the U.S. Naval Air Warfare Center, China Lake, Calif., has worked for several years using analog neural networks and the MRIII algorithm [2] in missile guidance and other military applications [26]. He has found that when fast decisions are required, neural networks have enormous advantages over conventional methods.

Fighter flight and battle pattern guidance. Defense contractors have apparently developed software using neural networks to integrate multi-source data for flight and battle pattern guidance of Lockheed's YF-22 Advanced Tactical Fighter based on real-time predictions of the imminent actions of an enemy aircraft. It is unclear, however, if such a system is operational [24].

Optical telescope focusing. Neural networks can be used to compensate for atmospheric disturbances by adaptively deforming mirror elements in response to atmospheric activity that can blur images. In strategic defense initiative-related work, Lockheed Missiles and Space Co. has developed a proprietary neural microchip that drives an adaptive focusing system for laser/mirror systems. This allows relatively small telescopes to rival much larger and more expensive ones. Colin Johnson reports in the November 19, 1990 issue of the *Electronic Engineering Times* that the first generation of the system had 69 piezoelectric actuators mounted on the back of the mirror to adjust it to the desired shape. Experiments with a similar idea utilizing a multiple mirror telescope are also described in the literature [22].

Vehicular trajectory control. Neural networks can be used to solve highly nonlinear control problems. A two-layer neural network containing 26 adaptive neural elements has learned to back up a computer-simulated trailer truck, even when initially "jackknifed." The neural net was able to learn of its own accord to do this, regardless of initial conditions. Experience gained with the truck backer-upper should be appli-

cable to a wide variety of nonlinear control problems [15].

Automotive applications. Ford Motor Co., General Motors, and other automobile manufacturers are currently researching the possibility of widespread use of neural networks in automobiles and in automobile production. Some of the areas that are yielding promising results in the laboratory include engine fault detection and diagnosis, antilock brake control, active-suspension control, and idle-speed control. General Motors is having preliminary success using neural networks to model subjective customer ratings of automobiles based on their dynamic characteristics to help engineers tailor vehicles to the market.

Electric motor failure prediction. Siemens has reportedly developed a neural network system that can accurately and inexpensively predict failure of large induction motors [26]. The system achieves 80% to 90% overall failure prediction accuracy in comparison to 30% achieved by the best conventional techniques. The predictor will be integrated into Siemens's existing Advanced Motor Master System (SAMMS) controller.

Speech recognition. The Stanford Research Institute (SRI) is currently involved in research combining neural networks with hidden Markov models (HMM) and other technologies in a highly successful speaker-independent speech recognition system. The technology will most likely be licensed to interested companies once perfected.

Mass spectra classification. Bo Curry of Hewlett-Packard Labs collaborated with David Rumelhart on the design of a feedforward neural network to classify low-resolution mass spectra of unknown compounds according to the presence or absence of 100 organic substructures. Described in HPL Technical Report 90-161, 1990, the neural network MSnet was trained to compute a maximum-likelihood estimate of the probability that each substructure is present. MSnet classifies mass spectra more reliably than other methods reported in the literature, is much faster than the standard nearest-neighbor techniques, and because of the probabilistic interpre-

*Many neural net applications are under development
in the **telecommunications industry**
for solving **control** problems.*



tation of the classification output, can readily be combined with other information sources.

Biomedical applications. Neural networks are rapidly finding diverse applications in the biomedical sciences. They are being used widely in research on amino acid sequencing in proteins, nucleotide sequencing in RNA and DNA, ECG and EEG waveform classification, prediction of patients' reactions to drug treatments, prevention of anesthesia-related accidents, arrhythmia recognition for implantable defibrillators, patient mortality predictions, quantitative cytology, detection of breast cancer from mammograms, modeling schizophrenia, clinical diagnosis of lower-back pain, enhancement and classification of medical images, lung nodule detection, diagnosis of hepatic masses, prediction of pulmonary embolism likelihood from ventilation-perfusion lung scans, and the study of interstitial lung disease.

Drug development. One particularly promising area of medical research involves the use of neural networks in predicting the medicinal properties of substances without expensive, time-consuming, and often inhumane animal testing [29]. For cancer drug screening, this has been accomplished by testing the effects that a group of 134 known drugs have on the growth of cultures of 60 types of human tumor cells. These profiles were then applied to a feedforward neural network simulated using NeuralWare's Professional II/PLUS software package and trained by backpropagation to classify each drug by mechanism of action. Cross-validation studies showed this method to be surprisingly accurate. The profiles of prospective drugs with unstudied medicinal properties could then be applied and classified by the network. More extensive tests would be performed only on the small proportion of prospective drugs placed by the network

in classes thought to be useful or interesting.

Control of copiers. The Ricoh Corp. has successfully employed neural learning techniques for control of several voltages in copiers in order to preserve uniform copy quality despite changes in temperature, humidity, time since last copy, time since change in toner cartridge, and other variables. These variables influence copy quality in highly nonlinear ways, which were learned through training of a backpropagation network. In order to improve generalization and reduce the size of the networks in copiers, Ricoh employed a sophisticated network-pruning method, which they call Optimal Brain Surgeon, which indeed led to smaller and more accurate networks.

More Detailed Descriptions of Selected Applications

The following subsections describe in greater depth a group of applications selected from the preceding summary. They all use some form of the delta rule or the backpropagation algorithm for adaptation and learning. The fields of application are highly diverse, but the learning processes are remarkably similar.

The telecommunications industry. Many neural network applications are under development in the telecommunications industry for solving problems ranging from control of a nationwide switching network to management of an entire telephone company. Other applications at the telephone circuit level turn out to be the most significant commercial applications of neural networks in the world today. Modems, commonly used for computer-to-computer communications and in every fax machine, have adaptive circuits for telephone line equalization and for echo cancellation. Adaptivity is needed because each telephone line has its own individual character-

istics, and these characteristics change over time.

Echo on telephone lines, which would normally be tolerated with speech, is devastating to high-speed data transmission. Echo cancelling solves the problem by detecting the echo and adding an equal and opposite signal to the return path. The cancelling signal is generated by an adaptive transversal filter whose coefficients (weights) are automatically adjusted by the LMS algorithm of Widrow and Hoff [32], also known as the delta rule in the field of neural networks. The adaptive filter makes use of what amounts to a single neuron. The first echo cancellers were developed at AT&T Bell Labs in the 1960s by M. M. Sondhi and his colleagues. Today they are everywhere.

The first application of adaptive techniques in telecommunications was telephone line equalization by Robert W. Lucky at AT&T Bell Labs. Telephone channels, radio channels, and even fiber-optic channels can have nonflat frequency responses and nonlinear phase responses in the signal passband. Sending digital data at high speed through these channels often results in a phenomenon called "intersymbol interference," caused by signal pulse smearing in the dispersive medium. Equalization in data modems combats this phenomenon by filtering incoming signals. A modem's adaptive filter, by adapting itself to become a channel inverse, can compensate for the irregularities in channel magnitude and phase response.

The adaptive equalizer in Figure 1 consists of a tapped delay line (a transversal filter) with a single adaptive neuron connected to the taps. Deconvolved signal pulses appear at the weighted sum, which is quantized to provide a binary output corresponding to the original binary data transmitted through the channel. The LMS algorithm is used to adapt the weights.

Figure 2a shows the analog response of a telephone channel carrying high-speed binary pulse data. Figure 2b shows an "eye" pattern, which is the same signal after going through a converged adaptive equalizer. Equalization opens the eye and allows clear separation of +1 and -1 binary data pulses.

Active control of sound and vibration. A new area of application for adaptive and learning systems to active control of noise and vibration, has been developing during the last 5 or 10 years. Passive control of noise would make use of thick walls and sound-absorbing materials and coatings, while passive control of vibration would make use of shock absorbers, damping materials and structures, and other methods of isolating and snubbing vibration. Active sound control uses adaptive techniques to generate antisound (equal and opposite) to cancel noise in a space or volume. Active vibration control uses adaptive techniques to

generate vibration to cancel existing vibration.

Active vibration control in a car is seen in the following example: Engine vibration coupling into the chassis through the four supporting engine mounts is cancelled by transducers shunting the engine mounts, which are driven so that equal and opposite forces are applied to the chassis. The transducer signals come from a set of adaptive filters, each utilizing a single neuron adapted by means of the "filtered-X" LMS algorithm [32].

Several companies have developed "electronic mufflers" which can replace the conventional passive mufflers in automobiles [23]. This is an example of active noise control. A tachometer on the engine generates pulses at the cylinder-firing rate. The tachometer signal is adaptively filtered, amplified, and fed to a small loudspeaker in the exhaust system. The loudspeaker generates antisound. The adaptive filter utilizes a single neuron that learns with the filtered-X LMS algorithm. The result is an engine that is at least as quiet as one with a conventional muffler. Additionally, the engine "breathes"

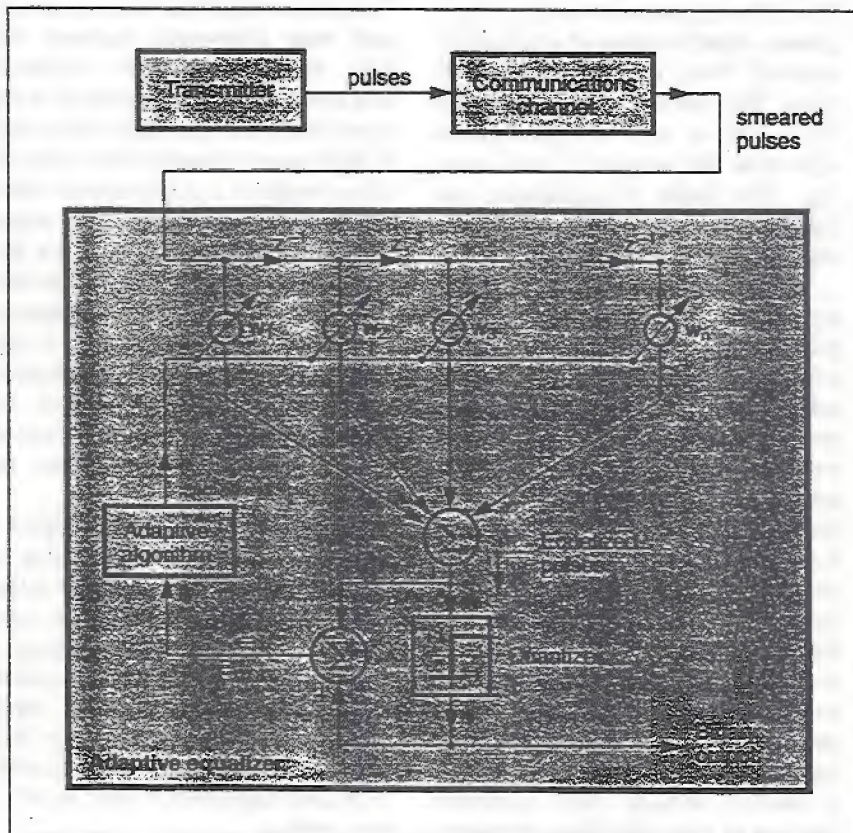
more easily, resulting in more horsepower and better fuel efficiency. As described by Randy Barrett in the August 12, 1993, issue of *Washington Technology*, Noise Cancellation Technologies (NCT) in a joint venture with Walker Manufacturing currently has electronic mufflers under test in New York City and Montreal bus fleets, where they have already demonstrated a 2.5% improvement in fuel economy. According to the October 28, 1992, issue of the *Electronic Engineering Times*, the first production vehicles with the NCT-Walker muffler should be available in 1996. A number of other automotive applications of the filtered-X LMS algorithm can be found in the proceedings of a conference on active control of sound and vibration held at Virginia Tech in April of 1991.

Active noise cancellation is also being developed to reduce noise problems caused by heating and air-conditioning equipment, vacuum cleaners, emergency vehicle sirens, aircraft, lawn mowers, and industrial equipment. NCT now markets a \$99 noise-cancelling headphone called NoiseBuster.

Beam control at the Stanford Linear Accelerator Center. The Stanford Linear Accelerator Center (SLAC) is a complex of particle accelerators operated by Stanford University for the U.S. Department of Energy. Physicists from all over the world design and perform experiments there, 24 hours a day, 7 days a week. A 3-kilometer-long linear accelerator fires both positrons and electrons into the circular arcs of a collider. A major challenge involves controlling the positions of the electron and positron beams in the collider to within 2 microns in spite of unpredictable disturbances that take place in the accelerator (due to changes in temperature, barometric pressure, vibration, sensor noise and so forth). Collisions must occur in order for the physicists to do their work, and the probability of collisions depends on the accuracy of positioning the opposing positron and electron beams.

The linear accelerator is divided into 20 sections. Each section has beam position sensors and control

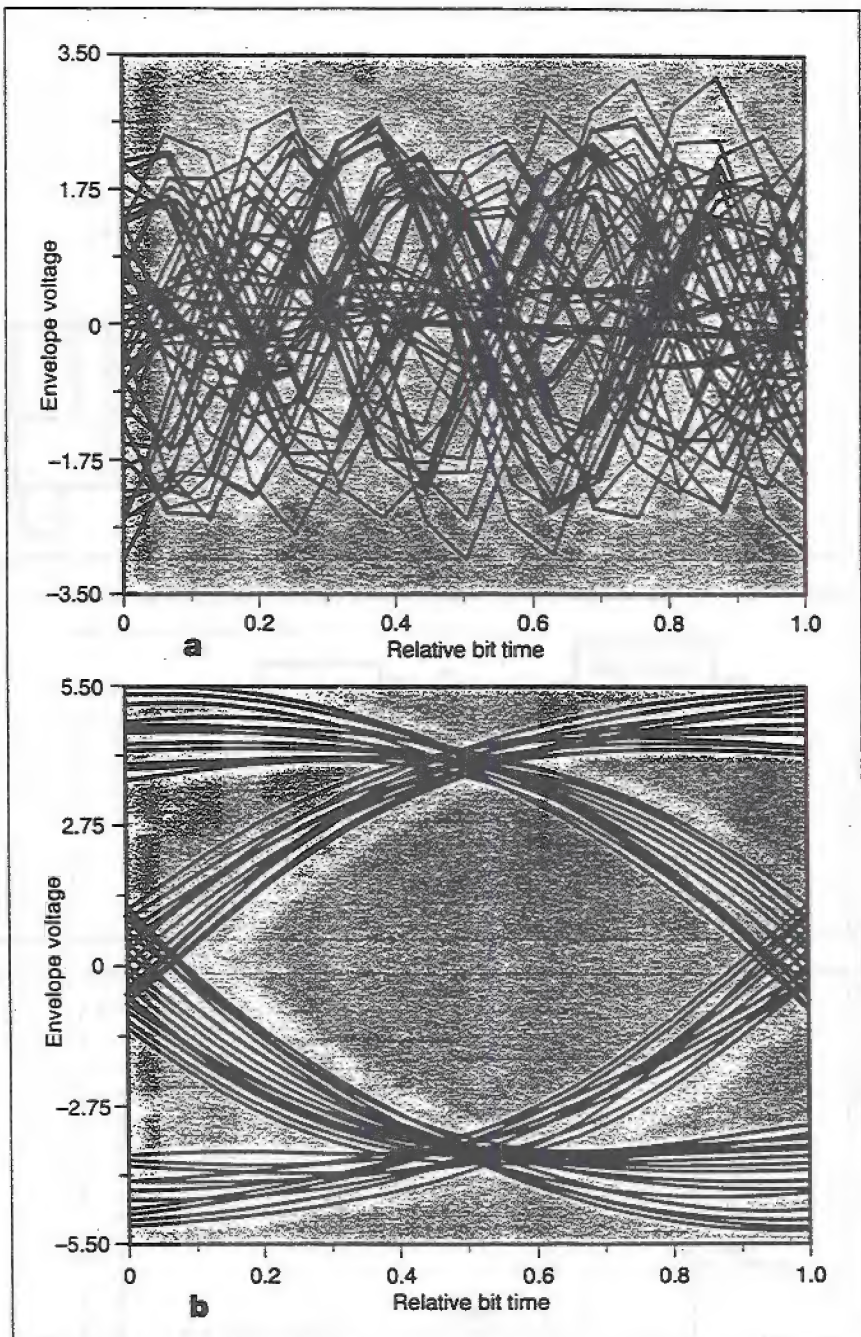
Figure 1. Adaptive channel equalizer with decision-directed learning



magnets to deflect the beam. Conventional feedback systems are used with each section for beam control, and they greatly reduce the variations in the beam position. Nonetheless, the system could not achieve the required accuracy without adaptive noise cancelling. Each section was equipped with a multi-input multi-output (MIMO) adaptive canceller, eight inputs, and eight outputs. This is equivalent to a neural network without nonlinearity. Adaptation was done by a MIMO form of the LMS algorithm. Prior to the installation of the new system, operators at the accelerator would frequently make frantic late-night phone calls for help in recovering from a problem. The system has been so robust and stable in the six months since the adaptive solution was implemented that the late-night phone calls have ceased, and no significant problems have occurred. (This work was performed by Thomas M. Himel of SLAC.)

The truck backer-upper. Vehicular control by artificial neural networks is a topic that has generated widespread interest. At Purdue University, tests have been performed using neural networks to control a model helicopter [16]. In a much larger project, a full-sized self-driving van named ALVINN (Autonomous Land Vehicle In a Neural Network) complete with video camera "eyes" and an onboard "brain" made from four workstations has been developed and built at Carnegie-Mellon University [18]. ALVINN learned to drive by watching humans drive and can drive long distances at normal highway speeds, negotiating through traffic without human intervention. The system is not yet perfect, of course, so when ALVINN drives, a human is always present to take over the controls if something goes wrong.

We now consider a system less complicated and more easily described than ALVINN—that of a neural network which has learned to steer a computer-simulated truck and trailer while backing to a loading platform. A solution to this highly nonlinear control problem was obtained by self-learning. The inputs to the two-layer network are "state" variables: the angle and position of



the rear of the trailer and the angle of the cab (see Figure 3). The output of the neural network is the angle of the steering wheel. The work was done by Nguyen and Widrow [15]. The learning algorithm they used, which is based on the famous backpropagation algorithm [21, 30, 31], is called backpropagation-through-time.

The truck was only allowed to back up. Backing was done as a sequence of small steps. On the scale of a real "18-wheeler," each step would be a distance of approximately one meter. The truck backs from its initial posi-

Figure 2. Eye patterns produced by overlaying cycles of the received waveform: a. before adaptive equalization; b. after adaptive equalization.

tion until it hits something and stops. The desired final state of the system involves having the rear of the trailer parallel to the loading platform and positioned at its center. The actual final state is compared with the desired final state, and the difference is a state error vector. After each backing-up sequence is completed, the final error vector is used to mod-

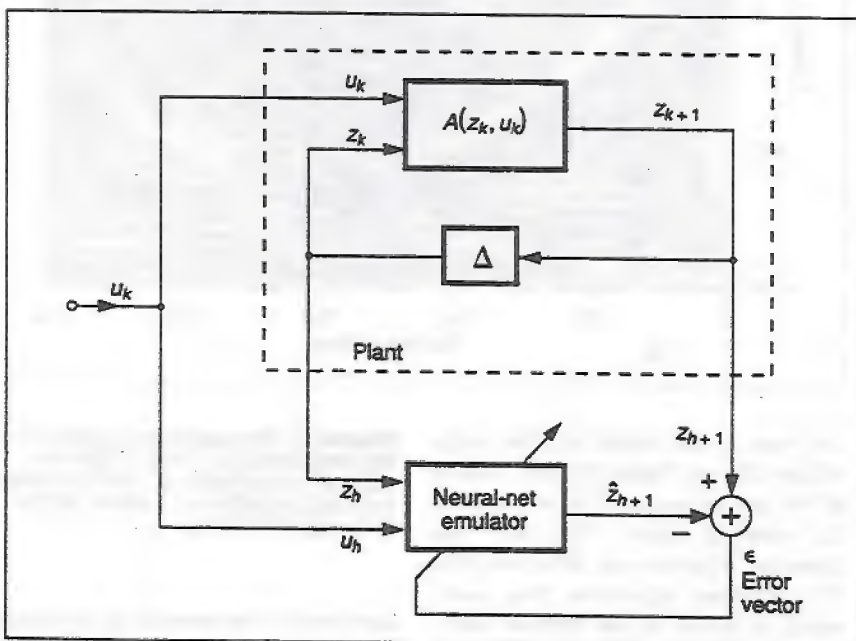
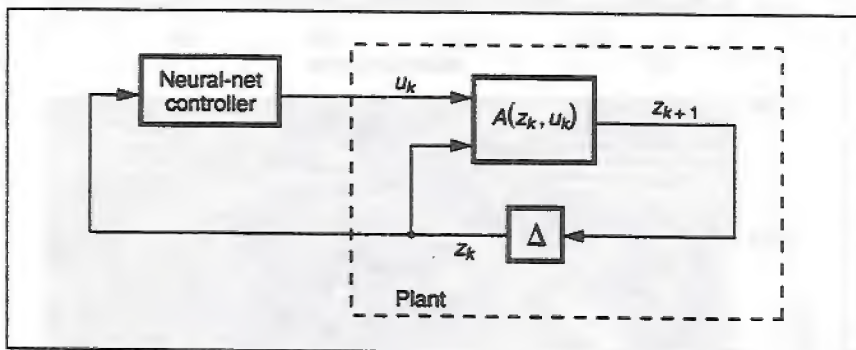
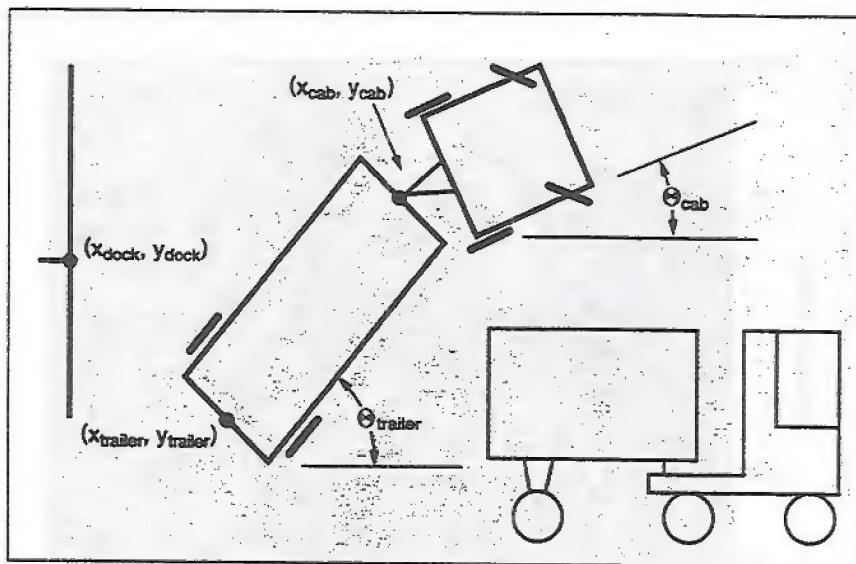


Figure 3. Truck, trailer, and loading dock

Figure 4. Plant and controller

Figure 5. Training the neural net plant emulator

ify the controller weights, so that if the truck is placed in the same initial position and allowed to retry the backup sequence, the new final-state error will have a smaller magnitude than before.

Figure 4 is a diagram of the neural

net controller steering the truck—a controller governing a “plant” represented by the truck kinematics. To train the controller, an emulator of the truck kinematics is needed. This is a two-layer neural network trained by backpropagation as shown in Figure 5 to produce the same output states as the plant when both the emulator and plant have the same driving function.

The controller is a two-layer neural network trained as shown in Figure 6. The initial position or state of the truck, z_0 , is applied to the controller, which generates a single output, the steering wheel angle. Using this steering signal, the truck backs up a step. The process of using the controller to set the steering angle, and then backing a step is repeated until either the truck hits something or the number of time steps exceeds a predetermined constant.

Backing from state to state is represented by signals going through the layers of a neural net. The controller and emulator are each composed of two layers of adaptive neurons. Every backing step corresponds to signals going through four layers. By “unrolling” the control system’s feedback loop, the whole backup sequence can thus be represented as the forward propagation through a giant feedforward neural network containing a number of layers equal to four times the number of time steps. In a process called backpropagation-through-time, the final-error vector is backpropagated through all the layers of this composite network.

After each backup sequence, the backpropagation-through-time algorithm finds a gradient of the squared positional error of the truck’s final state with respect to the weights of the controller. This gradient is used to update the controller’s weights by stochastic gradient descent.

Once learning is complete, the truck is able to back up satisfactorily from almost any initial position, even “jackknifed,” and even from initial positions that were not previously encountered during training. The controller’s ability to react and respond reasonably to new positions is an example of generalization. An illustration of the functioning of an already-trained system is shown in

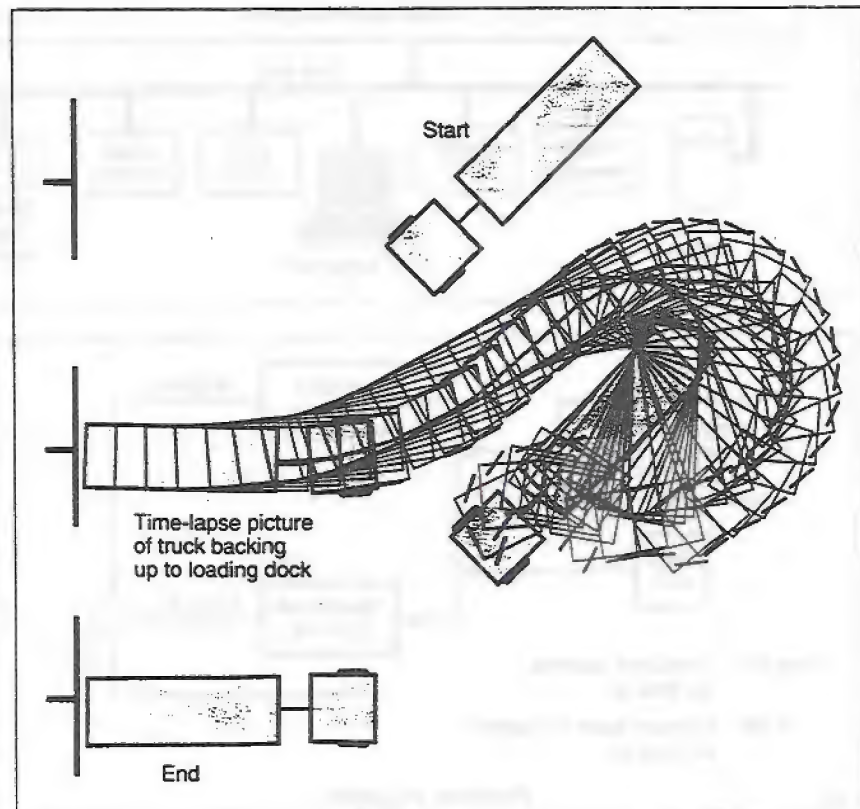
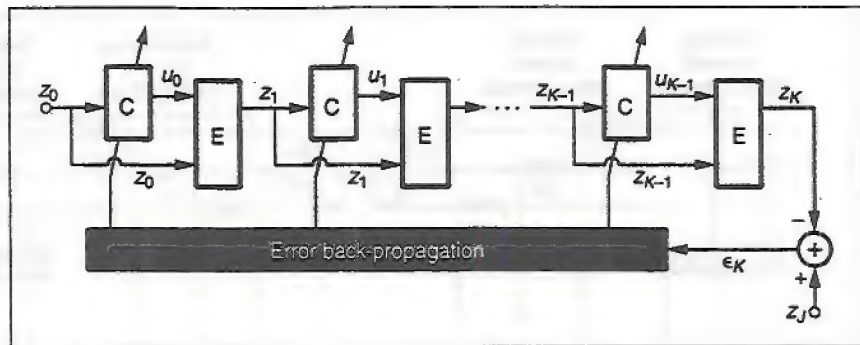
Figure 7. This is a laboratory exercise that could, in the future, have implications for vehicle control. Large American trucking companies are seriously exploring this technology. At the present time, the truck backer serves as a visual demonstration of the capabilities of nonlinear networks. This demonstration helped motivate development of the Intelligent Arc Furnace controller described next.

Steel making. An electric arc furnace is used to melt and process scrap steel. The heat energy comes from a three-phase power line of rather massive capacity (often 30 megawatts or more—enough electrical power for a city of 30,000 people). The three-phase line connects to a bank of step-down transformers to supply current for three electrodes that stick down into the furnace. The electrodes are made of graphite, are about one foot in diameter, and are about 20 feet long. Three independent servos control the depth of the electrodes into the furnace.

When starting a new "heat," scrap steel is loaded into the furnace, and the servos are activated to drive the electrodes down toward the scrap pile. When an arc is first struck, sparks fly, and the noise is deafening. One's first impression of this is that it is like Dante's inferno.

Because the cost of installing and operating a large arc furnace is so great, even small changes in efficiency have a tremendous impact on economics. The motivation for the development of "intelligent control" is clear. In this section we describe the Intelligent Arc Furnace controller, invented by Bill Staib of Neural Applications Corp. [28]. The figures in this section were supplied by the inventor.

Figure 8 shows an arc furnace, its three-phase power system, and instrumentation that provides signals useful for the control of the electrode servos. Currents and voltages in the system are sensed, digitized, and fed to a 486 PC that implements the neural control system. Numerical processing is performed by an 80-MFLOP Intel i860 microprocessor. A microphone placed near the furnace provides the computer with the



sounds of "Dante's inferno." From all the sensed variables, a state vector is obtained.

Figure 9a shows the training of a neural network emulator of the furnace. The idea is similar to that of Figure 5 for the truck backer. The emulator is used in the training of the controller or regulator, another neural network. Figure 9b shows the training of the regulator. The learning algorithm is a variant of the back-propagation algorithm. It works in a similar way to the training process for a single stage of Figure 6 of the truck backer.

The results with neural control thus far have been excellent compared with the control systems that commonly exist for arc furnaces. Consumption of electric power is

Figure 6. Training the controller with backpropagation (C = controller; E = emulator).

Figure 7. Example of a truck backup sequence

reduced by 5% to 8%; wear and tear on the furnace and the electrodes is reduced by about 20%; the power factor on the input power lines is brought closer to 1; and the daily throughput of steel is increased by 10%. The neural controllers are being installed by Neural Applications Corp. just as quickly as they can be produced. These improvements are reportedly worth millions of dollars per year per furnace.

The Chemical Process Industry Pavilion Technologies, Inc. of Aus-

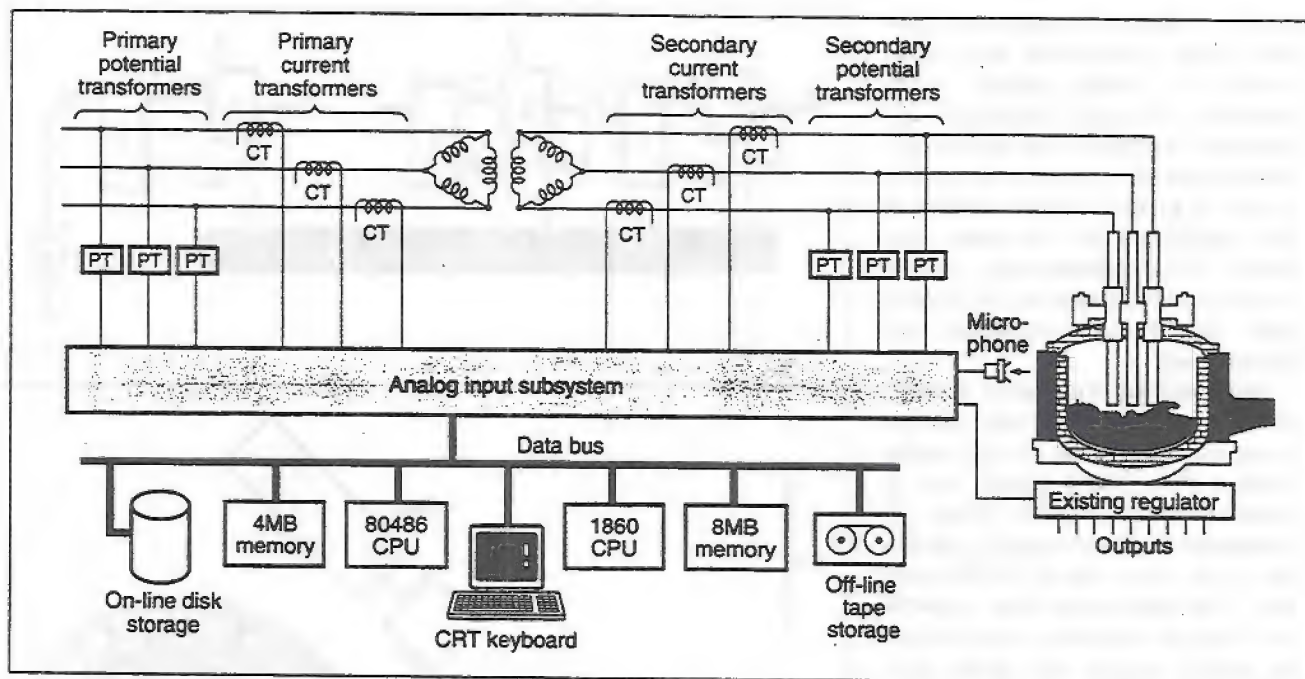


Figure 8. Arc furnace data acquisition system. Source: Courtesy of Bill Staib

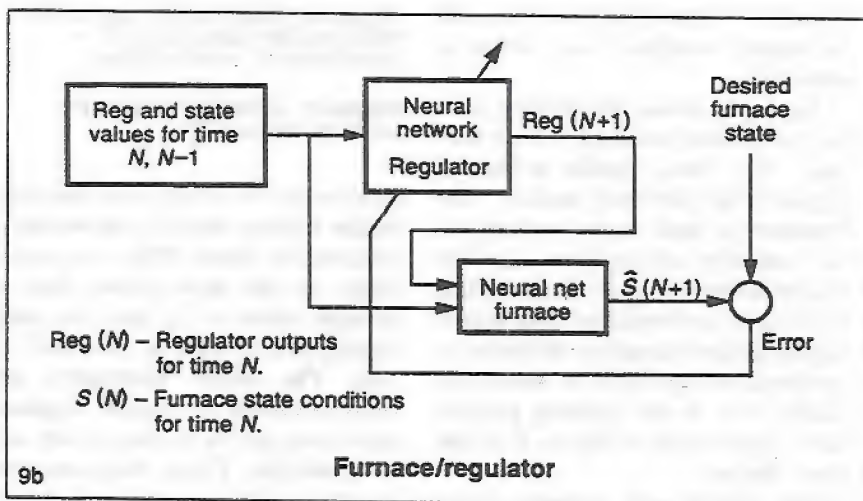
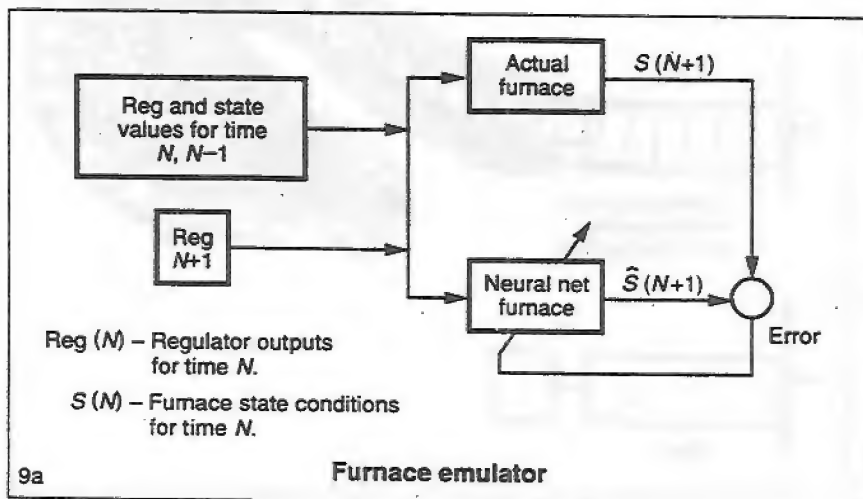


Figure 9. Block diagrams of a. furnace emulator; b. furnace/regulator. Source: Courtesy of Bill Staib

tin, Tex., has embedded neural networks and fuzzy logic into their Process Insights package for chemical manufacturing and control applications [4]. In this package, the user takes historical process data and uses it to build a predictive model of plant behavior. The model is then used to change the control setpoints in the plant to optimize behavior. Pavilion Technologies is a spin-off of MCC, where the original work was done in 1989 to 1990 by John Havener of Texas Eastman and Jim Keeler of MCC/Pavilion Technologies. In the original application conducted at the Texas Eastman Facility, Longview, Tex., neural networks in the Process Insights package produced setpoint changes that reduced by one-third the requirement of an expensive chemical additive needed to remove byproduct impurities during production. The facility produces plastics and chemical intermediates such as aldehydes and olefins. Since that work was completed, the technology and Pavilion's Process Insights software has been used in nearly 200 real-world applications, including modeling and optimization of distillation columns, modeling and control of plastics production, modeling and control of impurity levels in boil-

ers. These applications have generated tremendous paybacks, with savings of some applications totalling millions of dollars per year in single-unit production facilities. Texas Eastman, a division of Eastman Kodak, has been so satisfied with the results achieved by neural networks in the Process Insights package that they are currently encouraging the use of neural networks throughout their Longview plant. The success of the program is described in the April 29, 1993 issues of the company newsletter *Texas Eastman News*.

In making these applications, the first step is plant modeling or plant emulation. Typically, the plant has many inputs (such as pressures, temperatures, flow rates, and feed-stock characteristics) and one or more output parameters (such as yield, impurity levels, variance). In Figure 10 an

adaptive neural network is used to model an unknown plant (i.e., to learn the plant's dynamics from historical data).

Once the plant emulator converges, it can be used to train the neural net controller. Figure 11 shows how this is done. The error vector is the difference between the plant output vector and the desired-state vector. This error is backpropagated through the neural plant model to provide error signals for the adaptation of the weights of the controller. The controller weights are adapted by the backpropagation algorithm to minimize the sum of squares of the components of the error vector. Pavilion uses fuzzy logic in its Process Insights package to establish constraints on some of the controlled variables.

In most practical cases, it is not

possible to use a controller as simple as that shown in Figure 11. This is because almost all physical plants have internal dynamics. The plant's response to a control signal depends on both the current input to the plant and the current state of the plant. Any actions by the controller must therefore consider the state of the plant as well as its current input. A common solution involves incorporating tapped delay lines at the emulator and controller inputs to allow both networks to form internal representations of the present state. With tapped delay lines incorporated, Figure 11 then describes an increasingly popular form of open-loop control called nonlinear adaptive inverse control. Another approach is to incorporate one or more feedback loops in the system to create a dynamic system like the truck

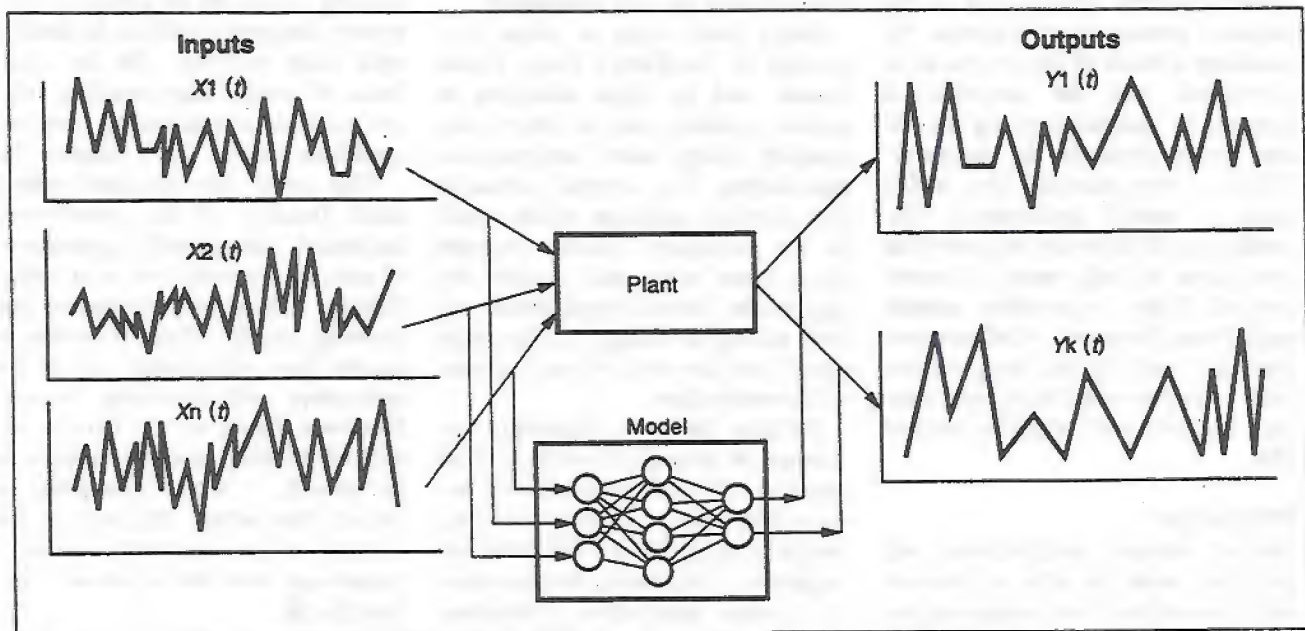


Figure 10. Adaptive plant emulation. Source: Courtesy of Jim Keeler

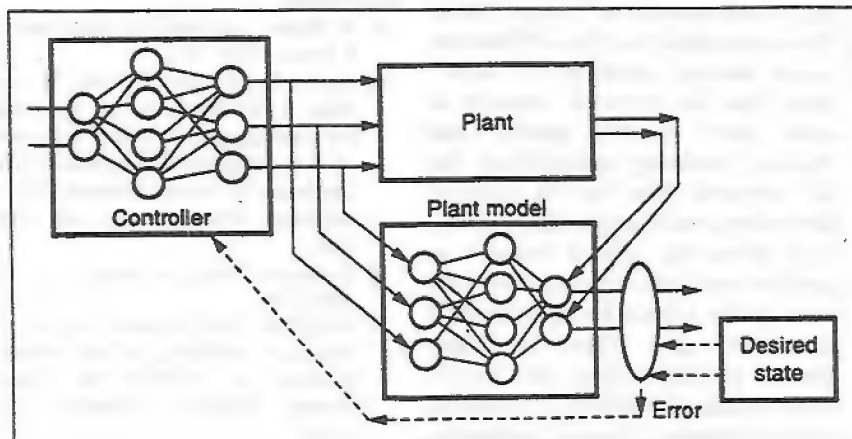


Figure 11. Using the plant model or emulator for backpropagation of error for training the neural controller. Source: Courtesy of Jim Keeler

backer. The controller can then be trained by backpropagation-through-time. Rather than simply using backpropagation to train the emulator as done with the truck backer, some closed-loop systems use backpropagation-through-time for this purpose as well [30].

In Process Insights, the relationship between the control history and the plant's state variables is determined by using measured states to train a dynamic state estimator [13]. The state estimator is then added to Figure 11 between the controller and the memoryless emulator. Memory is also added to the controller, which is trained by backpropagating error signals through the emulator and state estimator.

It is interesting to compare Figures 6, 9, and 11. Very similar things are going on in the vehicle control system (the truck backer), in the arc furnace control system, and in the chemical process control system. An emulator is made of the process to be controlled, and the controller is adapted by backpropagating the system error through the emulator. This is a very powerful idea, and it leads to useful applications. The reader should be aware however that this is not the only means of neural control. Other approaches include radial basis functions, reinforcement learning, and CMAC for problems such as process control, robotic actuator control, and vehicular control [34].

Conclusion

Neural network architectures will probably never be able to compete with conventional techniques at performing precise and well-defined numerical operations such as matrix inversions or Fourier transforms. However, there are large classes of problems that appear to be more amenable to solution by neural networks than by other available techniques. These tasks often involve ambiguity, such as that inherent in handwritten character recognition. Problems of this sort are difficult to tackle with conventional methods such as matched filtering or nearest-neighbor classification, in part because the metrics used by the brain to compare patterns may not be very

closely related to those chosen by an engineer designing a recognition system. Likewise, because reliable rules for recognizing a pattern are usually not at hand, fuzzy logic and expert system (ES) designers also face the difficult and sometimes impossible task of finding acceptable descriptions of the complex relations governing class inclusion. In trainable neural network systems, these relations are abstracted directly from training data. Moreover, because neural networks can be constructed with numbers of inputs and outputs ranging into the thousands, they can be used to attack problems that require consideration of more input variables than could be feasibly utilized by most other approaches. It should be noted, however, that neural networks will not work well at solving problems for which sufficiently large and general sets of training data are not obtainable.

Other tasks, such as those performed by VeriFone's Onyx Check Reader and by event detectors in particle colliders, can be solved successfully using more conventional approaches, but neural networks help provide solutions which result in less hardware. Faster response times, lower costs, and quicker design cycles. Several applications are now taking advantage of the high speeds and low costs of various neural network chips.

Perhaps the most important advantage of neural networks is their adaptivity. Neural networks can automatically adjust their parameters (weights) to optimize their behavior as pattern recognizers, decision makers, system controllers, predictors, and so forth. Self-optimization allows the neural network to "design" itself. The system designer first defines the neural network architecture, determines how the network connects to other parts of the system, and chooses a training methodology for the network. The neural network then adapts to the application. Adaptivity allows the neural network to perform well even when the environment or the system being controlled varies over time. There are many control problems that can benefit from continual nonlinear modeling and adaptation. Neural networks,

such as those used by Pavilion in chemical process control, and by Neural Applications Corp. in arc furnace control, are ideally suited to track problem solutions in changing environments. Additionally, with some "programmability," such as the choices regarding the number of neurons per layer and number of layers, a practitioner can use the same neural network in a wide variety of applications. Engineering time is thus saved.

Another example of the advantages of self-optimization is in the field of ES. In some cases, instead of obtaining a set of rules through interaction between an experienced expert and a knowledge engineer, a neural system can be trained with examples of expert behavior. The neural net becomes, in a sense, a trainable ES. Although it would implement rules, the actual rules implemented would not be apparent. The system designer would not be dealing with rules explicitly. On the other hand, if precise and complete rules are available or obtainable, then one would do best to use a classical ES.

This article has described only a small fraction of the commercial, industrial, and scientific applications of neural networks that exist today. The list is long and impressive and growing rapidly. There is no way to predict how widespread use of the technology will eventually become. However, based on the current extent of the field, and the rapidity of its growth, it seems reasonable to expect that before the turn of the century, neural networks will be a household word and a part of everyday life. □

References

1. *AI Expert*. Loudspeaker diagnosis. 5, 6 (June 1990), 71.
2. Andes, D., Widrow, B., Lehr, M., and Wan, E. MR11: A robust algorithm for training analog neural networks. In *Proceedings of the International Joint Conference on Neural Networks*. Vol. 1. Erlbaum, Hillsdale, N.J., pp. 533-536.
3. *Economist*. Tilting at chaos. (Aug. 15, 1992), 70.
4. Ferguson, R.B. Chemical process optimization utilizing neural network systems. In *SICHEM '92* (Seoul, Korea). Erlbaum, Hillsdale, N.J., 1992.

5. Flam, F. Neural nets: A new way to catch elusive particles. *Science* 256, 5061 (May 29, 1992), 1282-1283.
6. Fuochi, A. Neural networks: No zealots yet but progress being made. *Comput. Can.* 18, 2 (Jan. 20, 1992), 16.
7. Gill, T. and Shutt, J. Optimizing product formulations using neural networks. *Sci. Comput. Automat.* (Sept. 1992).
8. Hall, C. Neural net technology: Ready for prime time? *IEEE Expert* (Dec. 1992), 2-4.
9. Hammerstrom, D. Neural networks at work. *IEEE Spectr.* (June 1993), 26-32.
10. Hoffman, T. Don't choose technology without him. *Computerworld* 27, 4 (Jan. 25, 1993), 27.
11. Johnson, R.C. What is cognitive computing?: Intelligence from nature conquers tough programming problems. *Dr. Dobbs J.* 18, 2 (Feb. 1993), 18-22.
12. Kane, L.A. How combined technologies aid model-based control. *Hydrocarbon Proc.* (May 1993), 23.
13. Keeler, J.D. Prediction and control of chaotic chemical reactions via neural network models. In *Proceedings of the 1993 Conference on Artificial Intelligence in Petroleum Exploration and Production* (Plano, Tex., May 19-22, 1993).
14. Kestelyn, J. Neural net for quality control. *AI Exp.* 5, 10 (Oct. 1990), 71.
15. Nguyen, D. and Widrow, B. The truck backer-upper: An example of self-learning in neural networks. In *Proceedings of the International Joint Conference on Neural Networks*. Vol. II. Erlbaum, Hillsdale, N.J., pp. 357-363.
16. Pallett, T.J. and Ahmad, S. Real-time neural network control of a miniature helicopter in vertical flight. In *Proceedings of the Seventeenth International Conference on Applications of Artificial Intelligence in Engineering—AIENG/92* (Waterloo, Ontario, Canada, 1992).
17. PC AI. Product guide: Fuzzy logic—neural networks. (Mar./Apr. 1993), 52-56.
18. Port, O. Sure, it can drive, but how is it at changing tires. *Bus. Week* (Mar. 2, 1992), 98-99.
19. Rennie, J. Cancer catcher: Neural net catches errors that slip through pap tests. *Sci. Am.* 262, 5 (May 1990), 84.
20. Rumelhart, D.E. Theory to practice: A case study—recognizing cursive handwriting. In *Proceedings of the Third NEC Research Symposium*. SIAM, Philadelphia, Pa., 1993.
21. Rumelhart, D.E., Hinton, G.E., and Williams, R. J. Learning internal representations by error propagation. In *Parallel Distributed Processing*. Vol. 1. The MIT Press, Cambridge, Mass., 1986, chap. 8.
22. Sandler, D.G., Barrett, T.K., Palmer, D.A., Fugate, R.Q., and Wild, W.J. Use of a neural network to control an adaptive optics system for an astronomical telescope. *Nature* 351 (May 1991), 300-302.
23. Schuon, M. New technology chips away at noise. *New York Times* (Jan. 27, 1991). National Edition, sec. 1, p. 30.
24. Schwartz, E.I. Where neural networks are already at work: Putting AI to work in the markets. *Bus. Week* (Nov. 2, 1992), 136-137.
25. Schwartz, E.I. and Treece, J.B. Smart programs go to work: How applied-intelligence software makes decisions for the real world. *Bus. Week* (Mar. 2, 1992), 97-105.
26. Shandle, J. Neural networks are ready for prime time. *Elect. Des.* 41, 4 (Feb. 18, 1993), 51-58.
27. Shea, P.M. and Lin, V. Detection of explosives in checked airline baggage using an artificial neural system. In *Proceedings of the International Joint Conference on Neural Networks*. Vol. II (Washington, D.C., June 1989), pp. 31-34.
28. Staib, W.E. and Staib, R.B. The intelligence arc furnace controller: A neural network electrode position optimization system for the electric arc furnace. In the *International Joint Conference on Neural Networks*. IEEE, New York, 1992.
29. Weinstein, J.N., Kohn, K.W., Grever, M.R., Viswanadhan, V.N., Rubinstein, L.V., Monks, A.P., Scudiero, D.A., Welch, L., Koutsoukos, A.D., Chiausa, A.J., and Paull, K.D. Neural computing in cancer drug development: Predicting mechanism of action. *Science* 258, 16 (Oct. 1992), 447-451.
30. Werbos, P.J. Neural networks, system identification, and control in the chemical process industries. In *Handbook of Intelligent Control*. Van Nostrand Reinhold, New York, 1992.
31. Widrow, B. and Lehr, M.A. Thirty years of adaptive neural networks: Perceptron, madaline, and backpropagation. In *Proceedings of IEEE 78*, 9 (Sept. 1990), pp. 1415-1442.
32. Widrow, B. and Stearns, S.D. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1985.
33. White, D.A. and Sofge, D.A., Eds. *Handbook of Intelligent Control*. Van Nostrand Reinhold, New York, 1992.
34. Wright, D.P. and Scofield, C.L. Divide and conquer. *BYTE* (Apr. 1991), 207-210.

About the Authors:

BERNARD WIDROW is professor of electrical engineering at Stanford University. He does research and teaching in the fields of digital signal processing, adaptive signal processing, adaptive control systems, pattern recognition, and neural networks. He is coinventor of the LMS algorithm and the neural element ADALINE and various MADALINE networks.

DAVID E. RUMELHART is professor of psychology at Stanford University. His research has focused on how people learn complex skills such as reading, and how that knowledge is represented in the mind. He is coauthor of the well-known 2-volume set of connectionist texts, *Parallel Distributed Processing*.

MICHAEL A. LEHR is a doctoral candidate in electrical engineering at Stanford University. His research involves the application of second-order training techniques to large neural networks.

Authors' Present Addresses: Bernard Widrow and Michael Lehr can be reached at Stanford University Department of Electrical Engineering, Durand Bldg., Stanford, CA 94305-4055. David Rumelhart can be reached at Stanford University Department of Psychology, Bldg. 420 Room 414, Stanford, CA 94305-2130.

Support for this work was provided by the National Science Foundation under grant NSF IRI 91-12531, the ONR under contract no. N00014-92-J-1787, the EPRI under contract RP-8010-13, and the Department of the Army Belvoir RD&E Center under contract no. DAAK70-92-K0003. permission

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© ACM 0002-0782/94/0300 \$3.50

Interested in Standards?
Subscribe to ACM's
new publication...
StandardView
1-800-342-6626
(In U.S. & Canada)
1-212-626-0500
(In Metro N.Y. & Outside U.S.)

Genetic and Evolutionary Algorithms Come of Age

DAVID E. GOLDBERG

Before there were computers, there was thinking about the mind as a computer—as a machine. And in this way, computer science and engineering trace their roots to using natural examples. Within these fields of endeavor, AI drew its initial inspiration from nature, and work on computer-simulated brains received the lion's share of the early attention. But even back then, nature's other metaphor of adaptation planted a different seed that is now blossoming around the globe. Specifically, Darwinian evolution has spawned a family of computational methods called genetic algorithms (GAs) or evolutionary algorithms (EAs).

These search procedures, based on the mechanics of natural selection and genetics, are finding increasing application to difficult search, optimization, and machine-learning problems across a wide spectrum of human endeavor. Although for some years these investigations have remained cloistered in universities and research institutes, a new class of real-world applications is graduating from college and is moving into the computer rooms of industry and government, with repercussions starting to be felt from the factory floor to the community at large.

What Are Genetic Algorithms?

GAs¹ are search procedures based on natural selection and genetics. There are many variations on these algo-

rithms. For concrete discussion, we limit ourselves to the simple GA presented elsewhere [7], a GA that processes 'a finite population of fixed-length binary strings. In practice, bit codes, *k*-ary codes, real (floating-point) codes, permutation (order) codes, Lisp codes, and others have all been used with success. Each of these has their place, but here we examine a simple GA to better understand basic mechanics and principles.

A simple GA consists of three operators: selection, crossover, and mutation.

Selection is the survival of the fittest within the GA. There are many ways to achieve effective selection, including ranking, tournament, and proportionate schemes, but the key notion is to *give preference to better individuals*. For example, in two-party tournament selection, pairs of strings are drawn randomly from the parental population, and the better individual places an identical copy in the mating pool. If a whole population is

selected in this manner, each individual will participate in two tournaments and the best individual in the population will win both trials. The median individual will typically win one trial and the worst individual does not win at all.

Of course, for selection to function there must be some way of determining what is good. This evaluation can come from a formal objective function, or it can come from the subjective judgment of a human observer or critic. As the tournament selection example makes clear, the primary requirement is for a partial ordering.

If we were to do nothing but selection, GAs would not be very interesting because the trajectory of populations could contain nothing but changing proportions of strings contained in the original population. In fact, if run repeatedly, selection alone is a fairly expensive way of—with high probability—filling a population with the best structure of the initial population.

¹For simplicity, in remainder we will use the term GAs to mean either evolutionary or genetic algorithms. Historically, the word evolutionary has been associated with algorithms that use selection and mutation alone, while the term genetic has been associated with algorithms that use selection, mutation, recombination, and a variety of other nature-inspired mechanisms.

To do something more sensible, the algorithm needs to explore different structures. A primary exploration operator used in many GAs is crossover, and simple, one-point crossover proceeds in three steps. First, two individuals are chosen from the population using the selection operator, and these two structures are considered to be mated. A cross site along the string length is chosen uniformly at random, and position values are exchanged between the two strings following the cross site. For example, starting with the two strings $A = 11111$ and $B = 00000$, if the random choice of a cross site turns up a 3, we would obtain the two new strings $A' = 11100$ and $B' = 00011$ following crossover; these strings would be placed in the new population. This process continues, pair by pair, until the new population is complete, filled with "off-strings" that are constructed from the bits and pieces of good (selected) parents. There are many other variants of crossover, and many claims are made by their adherents. However, the main issue is whether the operator promotes the successful exchange of necessary substructures [10].

Selection and crossover are surprisingly simple operators, involving nothing more complex than random number generation, string copying, and partial string exchanges. Yet their combined action is responsible for much of a genetic algorithm's search punch. To understand this intuitively, we need only to think in terms of our own human processes of innovation. What is it we are doing when we are being innovative or creative? Often we are combination *notions* that worked well in one context with notions that worked well in another context to form new, possibly better *ideas* of how to attack the problem at hand [6]. Similarly, GAs juxtapose many different, highly fit substrings (notions) through the combined action of selection and crossover to form new strings (ideas).

If selection and crossover provide much of the innovative capability of a GA, what is the role of the mutation operator? In a binary-coded GA, mutation is the occasional (low-probability) alteration of a bit position, and with other codes a variety of

diversity-generating operators may be used. By itself, mutation induces a simple random walk through string space. When used with selection alone, the two combine to form a parallel, noise-tolerant hill-climbing algorithm. When used together with selection and crossover, mutation acts as both an insurance policy against losing needed diversity and as a hill climber.

Heated debates among evolutionaries and genetic algorithmists consider the relative importance of this operator or that, but most of this discussion is misplaced because GAs—and their natural counterparts—are *multifaceted*. Simple statements like "GAs are hill climbers," "mutation is the most important operator" or "crossover is the most important operator" are likely to be wrong, because GAs are complex systems, behaving differently in different portions of their phase space. The simplest GAs are discrete, nonlinear, stochastic, highly dimensional algorithms operating on problems of infinite variety. Not only does this get us into semantic difficulty, but because of this complexity, GAs are hard to design and analyze. However, just as the Wright brothers were able to design the complex system we now recognize as the airplane through an intuitive decomposition and the ruthless separation of subproblems [1], we, too, are able to design powerful GAs using a similar methodology of invention [5].

This design methodology relies heavily on Holland's notion of *schemata* and *building blocks* [11, 12]. Simply stated, schemata are similarity subsets (sets of strings that have one or more features in common), and building blocks are those schemata that are 1) consistently emphasized by selection and 2) respected and exchanged by the genetic operators. Since Holland's pioneering theories, much progress has been made in both experimentally verifying this *building-block hypothesis* and in following its design consequences. In particular, we appear on the verge of an integrated theory of simple GA operation [8, 10]. Moreover, one type of GA designed with strict adherence to building-block principles appears to give subquadratic results (in a prob-

ably approximately correct sense) to large ($l > 100$ bits) problems with billions of local optima [9]. More work is necessary to consolidate these findings and to spread them to the variety of codings in use, but the availability of well-grounded algorithms will prove important to practitioners as the stakes are raised and larger, more difficult applications are attempted.

Why Use GAs in Applications?

GAs can be attractive in applications work for a number of reasons:

1. GAs can solve hard problems quickly and reliably.
2. GAs are easy to interface to existing simulations and models.
3. GAs are extensible.
4. GAs are easy to hybridize.

One of the primary reasons to use GAs is that they are broadly competent algorithms. Empirical work has long suggested this, but theory is catching up, and it appears that GAs can solve problems that have many difficult-to-find optima. Moreover, because GAs work via sampling, populations may be sized to detect a given degree of function difference with no more than a specified amount of error [8]. This can make GAs remarkably noise tolerant.

Because GAs use very little problem-specific information, they are remarkably easy to connect to extant application code. Many algorithms require a high degree of interconnection between the solver and the objective function. For example, dynamic programming requires a stage-wise decomposition of the problem that not only limits its applicability, but can require massive rearrangement of system models and objective functions. GAs, on the other hand, have a clean interface, requiring no more than the ability to propose a solution and receive its evaluation. Oftentimes, getting a good model is nine-tenths of the battle, and once that model is tested and calibrated, the GA can be interfaced quite directly without additional difficulty. Moreover, because of a GA's noise tolerance, discrete-event simulations and other noisy evaluators can be used directly as long as population sizing is performed to account for the

stochastic variations in the evaluation process [8].

Even simple GAs can be broadly capable, but real problems can pose unanticipated difficulties. When these arise, oftentimes there is a solution from nature available to solve the problem. For example, in many AI problems the search spaces are highly multimodal, and the solution set may have multiple global solutions. In these cases, it is desirable to have the population converge to multiple optima simultaneously. In nature, of course, there is no superspecies that uses all resources everywhere on the planet, but instead there are multiple *species* that occupy multiple *niches*, separated from one another by the obstacles imposed by geography or through the utilization of different sets of resources. In a GA the notion of niche and species has been stably imposed on the population through various modifications to the selection scheme [7], and this kind of extensibility via nature is useful in GA design.

When nature does not call, it is often possible to use problem-specific information to help make a hybrid or knowledge-augmented GA. For example, many search domains have more competent local search heuristics than selection plus mutation, and getting the best answer in the shortest time often recommends combining the global perspective of the GA with the efficient local search of some problem-specific technique. There are also a number of ways that problem-specific information can be built into the operators or the codings, and a number of these are discussed in standard references [4, 7].

A Parade of Applications

For some time GAs were mainly an academic's plaything, but lately there have been an increasing number of industrial-strength applications gaining national and international attention. Here, we survey a sample of real-world GAs across a spectrum of problems from computer-aided engineering to finance, from criminal justice to fiber-optic network design. Although the codings, operators, and problem structure of the different applications are different, recurring themes will emerge, and we will visit these at the end of our march.

Products, Services, and Sources of Information

Genetic and evolutionary algorithms grew out of academic and research institutions, and today research activity is carried out at many locations, including the following: University of Alabama, University of Alberta (Canada), University of California at San Diego, Colorado State University, Dortmund University (Germany), George Mason University, University of Illinois at Urbana-Champaign, Kyoto University (Japan), University of Michigan, U.S. Naval Research Laboratory AI Center (Washington, D.C.), University of New Mexico, The Rowland Institute for Science, University of Tennessee-Knoxville, Tsukuba University (Japan), and Stanford University.

Commercially, a number of software packages are based on GA/EA technology:

- **Evolver**, in its second release from Axcels in Seattle, Wash., interfaces directly with Microsoft Excel to permit users to create an application model in the spreadsheet. In this mode, Evolver adjusts spreadsheet cells genetically, and objective function values are passed from Excel back to Evolver. In version 2.0, the GA is a .dll engine that can be accessed directly from other Microsoft Windows applications.
- **MicroGA** is available from Emergent Behavior in Palo Alto, Calif., as a library of C++ objects that implement a simple GA. Applications code must be written in C++ and interfaced with this library.
- **NeuralWorks Professional II/Plus**, from NeuralWare Inc. of Pittsburgh, Penn., has recently been outfitted with a Genetic Reinforcement Learning System. The system augments standard network training procedures by using a simple GA to avoid getting stuck at local optima.

Additionally, a number of private consultants specialize in GA applications. Tica Associates in Cambridge, Mass., was started in 1990 by Lawrence Davis as a consultancy specializing in the application of GAs. Boit, Beranek and Newman, Inc. (BBN), devotes considerable efforts on governmental and industrial applications of GAs, particularly in the areas of scheduling and military applications.

For those who would prefer to do their own hacking, a number of public-domain codes are available. Three books [4,7,14] contain sample codes that are readily available. Additionally, NASA's software distribution service COSMIC distributes a windows-oriented code called Splicer, which was developed through the Johnson Space Center by Mitre Corporation, and a GA written in C is available from the University of California at San Diego by contacting the Internet address nici@cs.ucsd.edu.

Started in 1985, the International Conference on Genetic Algorithms (ICGA) is the longest-running GA/EA conference, and it is held in odd-numbered years. A European conference, started in 1990, called Parallel Problem Solving from Nature is held in even-numbered years. Specialty conferences and workshops are popping up all over with the Workshop on the Foundations of Genetic Algorithms (FOGA, also in even-numbered years) being one of the most widely attended.

A number of journals devote considerable page space to GAs and EAs. *Adaptive Behavior* and *Evolutionary Computation* (both MIT Press) are two startup journals that consider GA/EA-related topics. *Complex Systems* contains articles, largely of foundational concern, and *The Annals of Mathematics and Artificial Intelligence* has published one special edition devoted to GAs and another is in the works. Two newsletters have followed GAs fairly closely: *Advanced Technology for Developers*, edited by Jane Kilmasauskas at NeuralWare, Pittsburgh, Penn., and *Release 1.0*, edited by Esther Dyson, New York. Electronically, the GA list is the oldest and most widely read GA-related news list with over 1,800 recipients (subscribe at ga-list-request@sun0.alc.nrl.navy.mil). Given the fast pace of GA developments, newsletters and electronic lists are almost essential to keep up with the latest.

Everyone Loves (a) CAD

One of the first major commercial applications came together in General Electric's computer-aided design (CAD) system EnGENEous [16]. This system was designed to be a domain-independent tool, combining the speedy local search of a number of traditional (and local) numerical optimization tools, the convenience of expert systems for specifying design constraints and control information, and the more global perspective of a genetic algorithm. The hybrid (or "interdigitized" in GE lingo) system can be interfaced to coordinate the activities of one or more domain-specific simulation or modeling codes, things as diverse as finite-element models, computational fluid dynamics codes, and discrete-event simulators.

EnGENEous got its start in engine design, evolving from an expert system code called Engineous. An early test on a 100-variable portion of a larger high-bypass gas turbine design problem compared human performance, the expert system code alone, the GA alone, the GA initialized by the expert system, and the interdigitized GA and expert system. All computer systems performed better than a human designer, with EnGENEous in interdigitized mode obtaining a 0.92% increase in turbine efficiency over the human designer, who was only able to get 0.5% better than the starting design. To the uninitiated these numbers might seem small, but improvements in a mature field like gas turbine design are hard fought, and even modest gains in efficiency translate into real customer savings and a significant competitive advantage for GE. An interesting aspect of the study was the effect of hybridization on computation time. The GA alone obtained a good solution, but required 30,000 function evaluations. When the GA was partially initialized using several expert system-derived designs, the time to good solutions dropped almost by a factor of five to 6,600 function evaluations. When the GA and the expert system were run iteratively the computational effort dropped to approximately 3,600 function evaluations. In all cases, the combined systems performed better than either pure system acting alone.

Since those early tests, EnGENEous has been updated to bring numerical optimization into the stable of tools used in the design process, and the system has been used successfully in a number of application areas. The gas turbine application has gone on to make the new Boeing 777 jet engine more efficient, and applications in electric-utility planning, hydroelectric generator design, and steam turbine design have been paying off handsomely. The latter application has been particularly important to GE, because steam turbines are designed and built on a custom basis with manufacturing times as short as 12 months. This constrains the design schedule to times as short as 2-3 weeks, and EnGENEous makes it possible to design more efficient nozzles, buckets, and other components through the integration of flow and resistance computer codes into the optimization process. In the past, such short lead times would only permit the examination of a few alternatives before a design was sent off to be manufactured.

Never Forget a Face

An application that is only a stone's throw—or a police station—away from the real world is the *Faceprints* system developed in the psychology department of New Mexico State University (NMSU) [2]. It was once common for police artists to draw a suspect's face from a witness's description, and more recently transparency-based sets of facial features have been used for the same purpose. These systems and straightforward computer implementations of them depend on the witness's ability to juxtapose individual eyes, mouths, and other facial features, but such a search process depends on the mind's ability to *recall* facial features individually, sometimes out of context. It is much easier for the mind to *recognize* similarity in a whole face that is close.

The NMSU system taps into the mind's eye by having a GA generate 20 faces on a computer screen. The witness rates each face on a 10-point subjective scale, and the GA takes that information and through normal selection, crossover and mutation, operators generate additional faces. The faces are generated from an underly-

ing binary chromosome that maps subcodes for each of five facial features—mouth, hair, eyes, nose, chin—into their pictorial representation, and the picture is assembled and displayed.

In testing the system, the NMSU team exposed subjects to a simulated crime and then asked them to use the system to reconstruct the criminal's face at varying times following the simulated crime. Figure 1 shows one result of one trial where a witness reconstructed the face three days following the simulated crime. The success of the technique has led NMSU to apply for a patent, and refinements to the technique have almost reached the point of commercialization.

Big Bucks from Yen (or Pounds or Marks)

Various AI systems have been used in financial applications, and GAs are no exception. A startup in Santa Fe, N.M., called the Prediction Company, has developed a set of time-series prediction and trading tools for currency trading in which GAs play an important role.

In particular, rule-like structures are evolved that have left-hand sides that are matched when time-series data enter specified regions [15], and the right-hand sides predict whether the time-series will go up or down. A collection of these rule structures form a population, and these are trained against real financial data, using an objective function that tries to reduce the mean-squared prediction error as it tries to increase the confidence of prediction.

In financial circles, one measure of investing efficacy is the Sharpe ratio, roughly the ratio of return to risk. In one currency-trading application, a known group of 20 currency traders was found to have Sharpe ratios in the range 0.3-1.0 with most traders in the range 0.3-0.7. Tests with the Prediction Company's technique demonstrated ratios as good as the best of the known currency traders. Recently, O'Conner Associates, a Chicago-based affiliate of Swiss Bank Corporation, entered into a long-term agreement to provide financing for Prediction Company's operations and trading.

Poetry to Our Ears

Interoffice fiber-optic networks are already a big business at US WEST, but an evolutionary algorithm developed in the Operations Research Modeling group [3] promises to make network additions faster and cheaper.

SONETs (synchronous optic networks) involve multiple rings of interconnected fiber-optic cable, where no more than 48 nodes are permitted per ring. Small networks at US WEST have been designed using traditional operations research techniques, but the design of the expansion of larger networks has been impractical by such methods. Design of such large networks was done by hand and relied on the designer's intuition and experience. As an intermediate step, these efforts were augmented by the use of commercially available network simulation tools. More recently, the evolutionary algorithm developed at US WEST has paid off dividends by allowing large networks to be designed efficiently and quickly. The specific technique uses a relatively small population with mutation-like operators that either expand an existing ring or start a new ring. The new alternative network is evaluated by running multi-period simulations on the commercial code. Networks that meet performance constraints are then compared on the basis of cost, with better networks surviving and worse networks dying off.

The tool was first tried in May 1992, and network design time has been cut from two person-months to roughly two person-days. Cost savings are estimated in the range from \$1 million to \$10 million per design, and with 20 designs required over the next six to eight years, total savings could top the \$100 million mark.

Other Applications

Space prohibits the fuller exploration of many of the interesting applications that are making or will soon make their mark. Here, we briefly survey a potpourri of applications in a number of different areas.

Keeping simulated top guns on time. Bolt, Beranek and Newman Inc. (BBN), of Cambridge, Mass., has created a GA-based scheduling algo-



Figure 1. The Faceprints system evolves a face through interaction with the witness to a crime. The images shown here are from laboratory tests with human subjects. On the left is the actual simulated criminal, and on the right is the Faceprints-generated image constructed three days after the simulated crime.

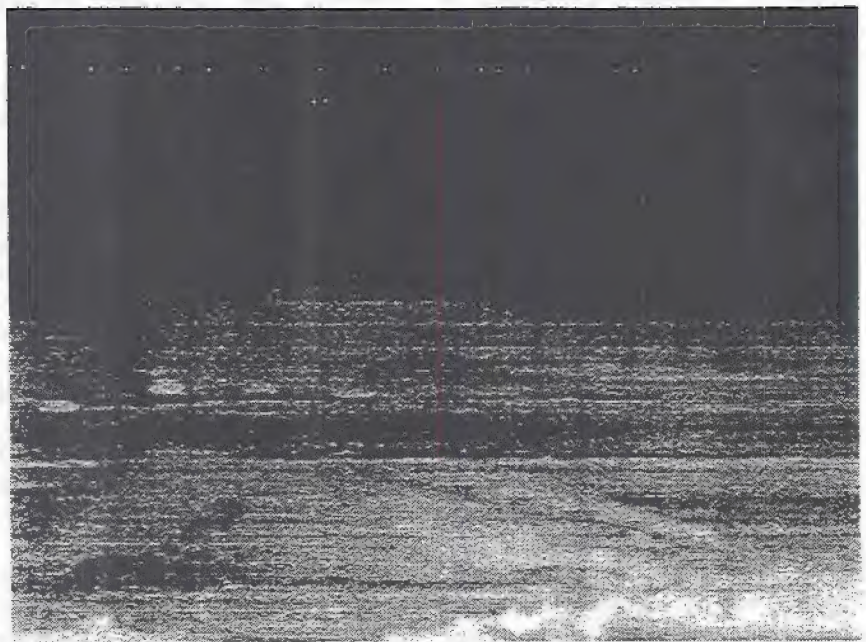


Figure 2. Eight tanks are shown in light clutter in a typical infrared image. The Hughes genetic-programming system is able to generate good tank detectors by recombining standard arithmetic operators and sets of primitive features.

rithm for the two System Integration Test Station Laboratories at the U.S. Navy's Point Mugu Naval Airbase [4]. These laboratories provide simulated environments for F-14 equipment and software testing, and the volume of testing demands every hour be used to the fullest. The BBN scheduler has been in use for a year and a half, and results have been good. The automatic system replaces a retiring human scheduler, and it captures the hard and soft constraints that were used in the manual scheduling process.

Tanks a lot. Hughes Missile Systems Company, Canoga Park, Calif., is applying the idea of genetic programming [14] to infrared image

target discrimination. In genetic programming, restricted Lisp expressions form the chromosome, and tree-based crossover recombines different structures while selection chooses the better ones. In the Hughes system [18], code composed of arithmetic operators and a logical-if operator are applied to inputs of 20 terminals, consisting of statistical features of a potential target. If the expression evaluates to a real value greater than zero, the image segment is taken to be a target. In tests against difficult data similar to that of Figure 2, genetic programming beat a neural network trained with backpropagation learning and a binary-tree classifier. The project has been so successful it is being implemented in hardware.

Fuzzy genetics. The Tuscaloosa office of the U.S. Bureau of Mines is moving a GA-adapted, fuzzy logic controller (GA-FLC) from the laboratory to the shop floor in a mineral recovery application. Off-line experiments [13] have shown that GA-FLCs are able to control pH and other relevant variables better than standard control algorithms to the point where industrial trials are warranted. Initial installation will begin shortly at Cliffs Mining Company in Ishpeming, Mich., and Kennecott Mining in Salt Lake City, Utah, with improvements in recovery efficiency and consistency expected shortly thereafter.

Let's get geophysical. Solving geophysical inverse problems is essential in oil exploration, and Advance Geophysical, Englewood, Col., is using a hybrid GA and gradient descent to solve what is called the *static correction problem* [17]. In seismic surveys, one of the first and most important corrections that is made to observed data accounts for the effect of reflection travel times off irregularities in the surface topography and the thickness and travel times in the low-velocity or weathered layer. In this application, the hybrid GA-gradient searcher is made available as part of a larger seismic-survey software package.

This quick applications rundown reveals a surprising breadth of application area as well as the use of different codings, operators, and objective functions. On the other hand, the applications are surprisingly similar

in their underlying motivation and approach. Many of the applications demonstrated a fairly rigid separation between model and searcher, and this is likely to be the case in many other applications as well. Also, many of the applications had useful heuristics and local search techniques and found it was useful to bring those on board to improve convergence times. Perhaps most importantly, each of the applications came to GAs for performance, not for fun. Although academic studies can afford the luxury of using technology for its own sake, practitioners cannot, but even with a nuts-and-bolts attitude like this, more and more applications are turning to GAs to solve hard problems that have long awaited computer solution.

Crystal Ball Not Needed

With an increasing number of practical applications in existence, the future of GAs seems fairly bright. Although this article has concentrated largely on activities in North America, interest and activity is strong in Europe (particularly the UK and Germany) and Japan. On a recent trip to Japan, I witnessed research and development activities in a number of major corporations, including NEC, Mitsubishi Electric, and Fujitsu, and I heard widespread rumors of patent and preemptive trademarking activity for GA-based products.

All applications efforts are aided by a growing body of practical theory that helps us understand what GAs process, how they can process it better, and how long and how close we should expect to come to global or near-global solutions in difficult problems. Tests on large-scale problems with billions of local optima are showing us that GAs can solve problems much harder than was once thought, and the convergence can be obtained more quickly and more reliably than was previously possible.

With practical successes growing in number and with theoretical results paving the way for practical, yet well-grounded applications, nature's favorite search algorithm may soon become industry's as well. It is becoming clear that GAs and EAs are changing our vision of what it is possible to design and operate. As knowl-

edge of this new intellectual leverage becomes more widespread, no crystal balls are needed to suggest that the fuller realization of robust computer aids leaves us standing on the threshold of what might soon be called a golden age of adaptation.

Acknowledgments

This article would not have been possible without the generous contributions of a number of individuals: Tony Cox at US WEST, Dave Davis at Tica Associates, Matt Jensen at Axcelis, Victor Johnston at New Mexico State University, Chuck Karr at the U. S. Bureau of Mines, Jane Klimasauskas at Neuralware, John Koza at Stanford University, Dave Montana and Gil Syswerda at BBN, Norman Packard at the Prediction Company, Christof Stork at Advance Geophysical Corp., Walter Tackett at Hughes Missile Systems, and Stewart Wilson at the Rowland Institute for Science. □

References

1. Bradshaw, G.L. and Lienert, M. The invention of the airplane. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Cognitive Science Society, 1991, pp. 605-610.
2. Caldwell, C. and Johnston, V.S. Tracking a criminal suspect through face-space with a genetic algorithm. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. International Society for Genetic Algorithms, 1991, pp. 416-421.
3. Cox, L.A., Kuehner, W.E., Parrish, S.H. and Qiu, Y. Optimal expansion of fiber-optic telecommunications networks in metropolitan areas. *Interf.* 23. To be published.
4. Davis, L., Ed. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
5. Goldberg, D.E. Making genetic algorithms fly: A lesson from the Wright brothers. *Adv. Tech. Devel.* 2 (Feb. 1993), 1-8.
6. Goldberg, D.E. Genetic algorithms as a computational theory of conceptual design. In *Applications of Artificial Intelligence in Engineering*. Vol. 6, 1991, pp. 3-16.
7. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass., 1989.
8. Goldberg, D.E., Deb, K. and Clark, J.H. Genetic algorithms, noise, and

- the sizing of populations. *Complex Syst.* 6, 8 (Aug. 1992), 333-362.
9. Goldberg, D.E., Deb, K., Kargupta, H. and Harik, G. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. IlliGAL Rep. No. 93004. Illinois Genetic Algorithms Lab., Univ. of Illinois at Urbana-Champaign, Ill., 1993.
 10. Goldberg, D.E., Deb, K. and Thierens, D. Toward a better understanding of mixing in genetic algorithms. *Soc. Instr. Contr. Eng. J.* 32, 1 (Jan. 1993), 10-16.
 11. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Cambridge, Mass., 1992.
 12. Holland, J.H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Mich.
 13. Karr, C.L. Design of an adaptive fuzzy logic controller using a genetic algorithm. In *Proceedings of the Fifth International Conference on Genetic Algorithms*. International Society for Genetic Algorithms, 1991, pp. 450-457.
 14. Koza, J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, Mass., 1992.
 15. Packard, N. A genetic learning algorithm for the analysis of complex data. *Complex Sys.* 4 (1990), 573-586.
 16. Powell, D.J., Tong, S.S. and Skolnick, M.M. EnGENEous domain independent, machine learning for design optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*. International Society for Genetic Algorithms, 1989, pp. 151-159.
 17. Stork, C. and Kusuma, T. Hybrid genetic autostatics: A new approach for large amplitude statics with noisy data. 1993. Manuscript submitted for publication.
 18. Tackett, W.A. Genetic programming

for feature discovery and image discrimination. 1993. Manuscript submitted for publication.

About the Author:

DAVID E. GOLDBERG is professor of General Engineering at the University of Illinois at Urbana-Champaign. **Author's Present Address:** Department of General Engineering, University of Illinois at Urbana-Champaign, 117 Transportation Bldg., Urbana, IL 61801; email: deg@uiuc.edu

Support for this work was provided by the U.S. Army under contract DASG60-90-C-0153, and by the National Science Foundation under grant ECS-9022007.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© ACM 0002-0782/94/0300 \$3.50

STANDARDS... what you need to know



Topics include:

- Human-computer interaction
- Governments and standardization
- Internationalization and localization
- Users and Standardization
- Objects and objectivity • Open systems

Sign up for your subscription to ACM's new standards magazine today!



Member rate: Only \$25
Individual Nonmember rate \$50

Call 1-800-342-6626

(In metro NY or outside US and Canada:
Call 1-212-626-0500)

SVAP93

TAP INTO ACM'S CAREERLINE

A New
Member Benefit
from

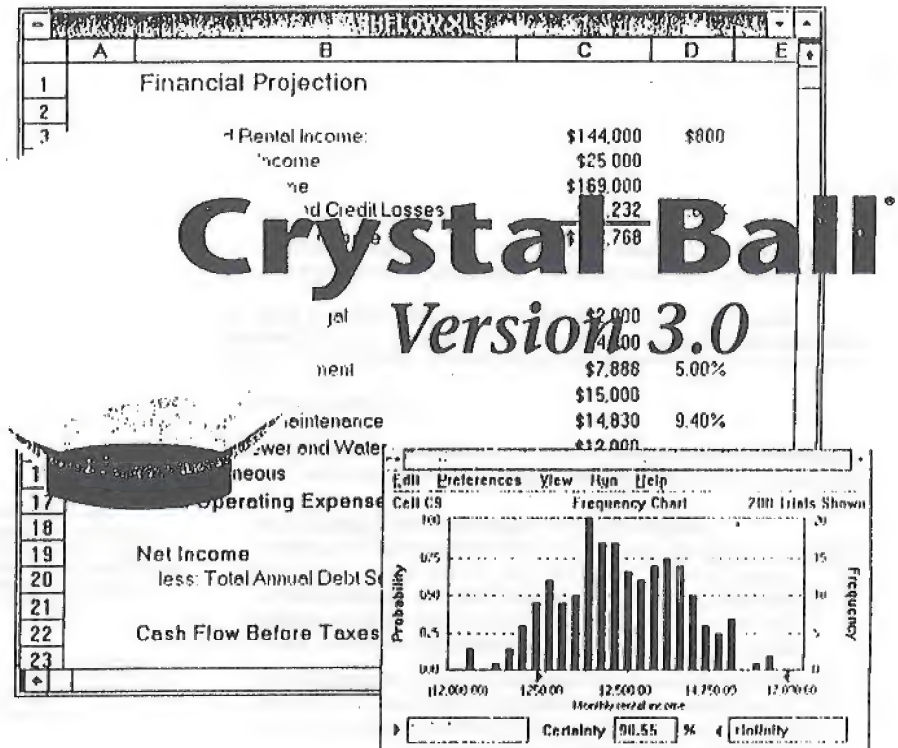


Looking for career advancement?
Contemplating a job or career change?
Worried about falling victim to a
corporate downsizing?

ACM's career counselling service can help with defining career goals, researching job opportunities, preparing a winning resume, soliciting and preparing for effective interviews, negotiating and evaluating job offers.

Contact: Jack Wilson, Career Sciences,
100 Mills Plains Road, Danbury CT 06811
Fax: (203) 431-8042. (bet. 5:30-8:30 EST)

Email Career@ACM.org. You must provide your Member number.



User Manual

DECISIONEERING

In this Chapter

- What Crystal Ball Does
- How Crystal Ball uses Monte Carlo Simulation
- "Futura Apartments" Spreadsheet Tutorial
- "Vislon Research" Spreadsheet Tutorial
- Defining Assumptions
- Selecting and Defining Distributions
- Running a Simulation
- Interpreting the Results

In this chapter are two tutorials, one short, one long, providing an overview of Crystal Ball's features. The first tutorial, the "Futura Apartments" spreadsheet, simulates profit/loss projections from apartment rentals. This tutorial is ready to run so you can quickly see how Crystal Ball works. If you work regularly with statistics and forecasting techniques, this may be all the introduction you need before running your own spreadsheets with Crystal Ball.

The second tutorial, the "Vislon Research" spreadsheet, gives you a chance to enter data and set up a complete simulation for a major corporate expenditure decision. As you work through the second tutorial, do not worry about making mistakes; recovery is as easy as backing up and repeating the steps. If you need additional help, refer to Appendix A, *Error Recovery*, or the Crystal Ball Help menus.

Now, spend a few moments learning how Crystal Ball can help you make better decisions under conditions of uncertainty.

What Crystal Ball Does

Glossary Term:
Spreadsheet Model - Any spreadsheet that represents an actual or hypothetical system or set of relationships.

Crystal Ball extends the forecasting capability of your spreadsheet model and provides the information you need to become a more accurate, efficient and confident decision-maker. As a spreadsheet user, you know that spreadsheets have two major limitations:

- You can only change one spreadsheet cell at a time. As a result, exploring the entire range of possible outcomes is next to impossible; you cannot realistically determine the amount of risk that is impacting your bottom line.
- "What-if" analysis always results in single-point estimates which do not indicate the likelihood of achieving any particular outcome. While single-point estimates may tell you what is *possible*, they will not tell you what is *probable*.

Crystal Ball overcomes both of these limitations and takes the guesswork out of spreadsheet analysis by providing fast and accurate results in minutes:

- You can describe a range of possible values for each uncertain cell in your spreadsheet. Everything you know about each assumption and how it affects your result is expressed all at once.
- Using a process called **Monte Carlo Simulation**, Crystal Ball displays your results in a forecast chart that shows the entire range of possible outcomes *and* the likelihood of achieving each of them. In effect, Crystal Ball moves you beyond "what-if" scenarios by providing an accurate statistical picture of the range of possibilities associated with your assumptions.

The best way to quickly understand this process is to start Crystal Ball and work on the first tutorial: the "Futura Apartments" spreadsheet.

1. Start Crystal Ball.

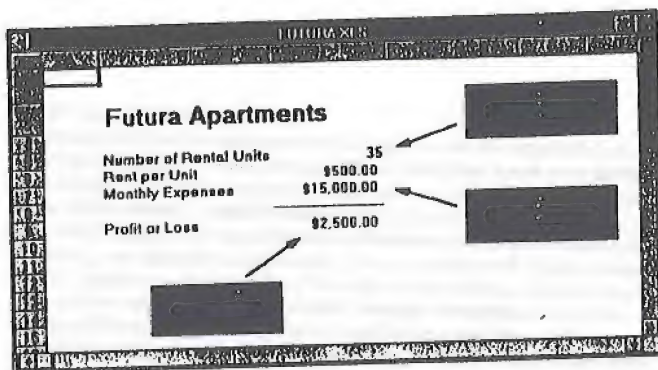
Crystal Ball Note: Directions for installing and starting Crystal Ball are in the "How to Install Crystal Ball" section at the front of this manual.

2. Open the "Futura Apartments" spreadsheet file, FUTURA.XLS (Excel) or FUTURA.WK4 (Lotus 1-2-3), from the Crystal Ball Examples directory.

Glossary Term:
Assumption - An estimated value or input to a spreadsheet model.

Glossary Term:
Monte Carlo Simulation - A system which uses random numbers to measure the effects of uncertainty in a spreadsheet model.

The "Futura Apartments" spreadsheet is displayed.



In this example, you are a potential purchaser of the Futura Apartments complex. You have researched the situation and created the above spreadsheet to help you make a knowledgeable decision. Your work has led you to make the following assumptions:

- \$500 per month is the going rent for the area.
- The number of units rented during any given month will be somewhere between 30 and 40.
- Operating costs will average around \$15,000 per month for the entire complex, but may vary slightly from month to month.

Based on these assumptions, you want to know how profitable the complex will be for various combinations of rented units and operating costs. As useful as spreadsheets are, this would be difficult to determine using a spreadsheet alone. The last two assumptions cannot be reduced to single values as required by the spreadsheet format. You would need to spend a great deal of time working through "what-if" scenarios, entering single values and recording the results, to try out all the combinations. Even then, you would likely be left with a mountain of data instead of the overall profit and loss picture.

With Crystal Ball, this kind of analysis is easy.

1. Choose Run from the Run menu on the menu bar.

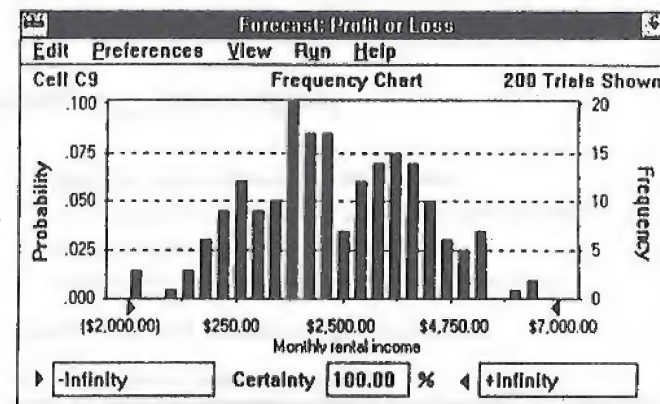
Crystal Ball runs a simulation for a situation in the "Futura Apartments" spreadsheet and displays a forecast chart as it is being created.

Glossary Term:
Iteration also Trial - A three-step process in which Crystal Ball generates random numbers for assumption cells, recalculates the spreadsheet model(s), and displays the results in a forecast chart.

After the simulation has run for at least 200 iterations, as displayed in your spreadsheet's status bar;

2. Choose Stop from the Run menu on the forecast chart (Excel) or on the menu bar above the forecast chart (Lotus 1-2-3).

Excel Note: If the forecast chart disappears behind Excel's window when a simulation is running, you can bring it back to the front by pressing Alt-Tab.



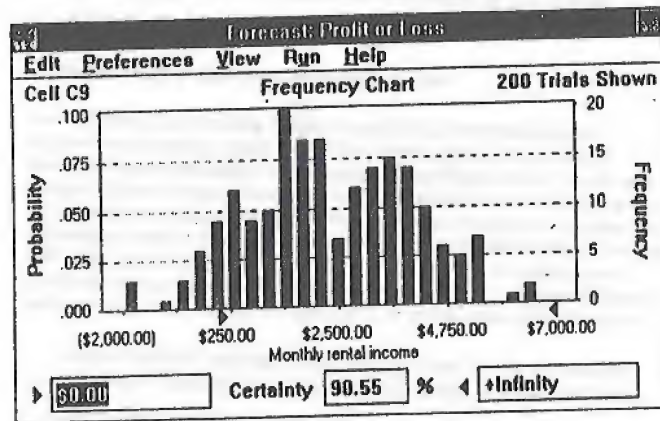
The forecast chart reveals the total range of profit and loss outcomes predicted for the "Futura Apartments" scenario. Each bar on the chart represents the likelihood, or probability, of earning a given income. The cluster of columns near the center indicates that the most likely income level is between \$250 and \$4,750 per month. Crystal Ball is also forecasting that the worst case is a \$2,000 loss and the best case is nearly a \$7,000 gain.

Now, you can use Crystal Ball to determine the statistical likelihood of making a profit.

1. Press the Tab key twice or select the left edit box on the forecast window.
2. Type 0 and press the Enter key.

Glossary Term:
Probability - (Classical Theory) the likelihood of an event occurring

The value in the Certainty box changes to reflect the probability of an income level ranging from \$0 to positive infinity—the probability of making a profit. With this information, you are in a much better position to make a decision on whether to purchase the Futura Apartments. In this case, there is 90.55% chance that you will make a profit, as shown below:



How Crystal Ball Uses Monte Carlo Simulation

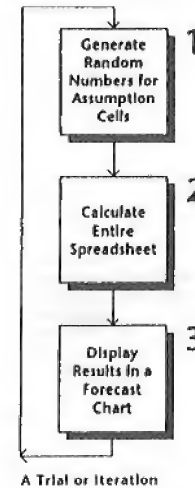
Most real-world problems involving elements of uncertainty are too complex to be solved by strict analytical methods. There are simply too many possible combinations of input values to calculate every possible answer.

Monte Carlo simulation is an efficient technique for analyzing these types of problems. It is a simple technique that requires only a random number table or a random number generator on a computer. Instead of calculating all possible combinations of input values, Monte Carlo simulation randomly chooses a relatively small number of values as inputs to the problem to generate a very good approximation of the answer. During a simulation, Crystal Ball uses Monte Carlo simulation to randomly create inputs to your problem that look like real-life possibilities, calculate the results, and plot them on a graph. Random numbers are distributed according to range estimates you define for the assumption cells.

Glossary Term:
Random Number - A mathematically selected value which is generated (by a formula or selected from a table) to conform to a probability distribution.

Glossary Term:
Random Number Generator - A method implemented in a computer program that is capable of producing a series of independent, random numbers.

The spreadsheet is recalculated to produce results for the forecast cells. Crystal Ball charts the forecast results in an easy-to-understand graphical format (forecast chart). As the numbers change in the assumption cells, the values in the forecast cells change, and the forecast chart displays these values.



Statistical Note: An Assumption is an input value. A Forecast is an output value.

This is an iterative process which continues until either:

- All of the trials specified for the simulation have been completed, or
- The simulation is stopped manually.

Keep in mind that the spreadsheet model can only approximate a real-world situation. When you build your own spreadsheet models, you will need to carefully examine your problem and continually refine the models until they reflect your real-world situation as closely as possible.

Crystal Ball also provides statistics that describe the forecast results. These are presented in detail in Chapter 2, *Understanding the Terminology*, and Chapter 4, *Interpreting the Results*.

When you are ready to move on to the next tutorial,

1. Choose Reset from the Run menu on the forecast chart.
2. Click OK in the dialog box that is displayed.
The simulation is reset and the forecast window disappears.
3. Close the "Futura Apartments" spreadsheet.

The "Vision Research" Spreadsheet

The remainder of this chapter contains a tutorial for the "Vision Research" spreadsheet. This tutorial provides a more realistic situation to let you examine Crystal Ball's features in greater depth. However, if you feel comfortable running Crystal Ball now, you may wish to turn to Chapter 2, *Understanding the Terminology*, for some background on Crystal Ball terminology. Then you can read Chapter 3, *Setting Up and Running a Simulation*, and start analyzing your own spreadsheets.

The "Vision Research" spreadsheet models a business situation filled with uncertainty. Vision Research has completed preliminary development of a new drug, code named ClearView, that is designed to correct nearsightedness. This revolutionary new product could be completely developed and tested in time for release next year if the FDA approves the product. Although the drug works well for some patients, the overall success rate is marginal, and Vision Research is uncertain whether the FDA will approve the product.

Vision Research will use Crystal Ball to help decide whether to scrap the project or proceed to develop and market this exciting new drug. The ClearView project is a multimillion dollar risk. Crystal Ball is a powerful decision-support program designed to take the mystery out of decisions like this.

To see how Crystal Ball works in a typical business decision:

1. Open the VISION.XLS (Excel) or VISION.WK4 (Lotus 1-2-3) spreadsheet from the Crystal Ball Examples subdirectory.

The "Vision Research" spreadsheet for the "ClearView Project" is displayed.

ClearView Project			Suggested Distributions:
Costs (in millions):			
Development Cost of ClearView to Date	\$10.0		Uniform Triangular
Testing Costs	\$4.0		
Marketing Costs	\$16.0		
Total Costs	\$30.0		
Drug Test (sample of 100 patients):			
Patients Cured	100		Binomial
FDA Approved if 20 or More Patients Cured	TRUE		
Market Study (in millions):			
Persons in U.S. with Nearsightedness Today	40.0		Custom
Growth Rate of Nearsightedness	2.00%		
Persons with Nearsightedness After One Year	40.0		

Take a look at the "Vision Research" spreadsheet on your screen. This spreadsheet models the problem that Vision Research is trying to solve. It includes value cells and formula cells. Value cells contain numeric values; formula cells contain formulas that refer to the value cells.

Defining Assumptions

Glossary Term:
Probability Distribution -
A set of all possible events
and their associated
probabilities.

In Crystal Ball, you define an assumption for a value cell by choosing a probability distribution that describes the uncertainty of the data in that cell. You select from the 16 distribution types in the Distribution Gallery.

How do you know which distribution type to choose? This portion of the tutorial has been designed to help you understand how to select a distribution type based on the answer you are looking for. In the following examples, you will select the assumption cells in the "Vision Research" spreadsheet and choose probability distributions that most accurately describe the uncertainties of the ClearView project.

This tutorial also explains the reasons for choosing a particular distribution for each assumption. Detailed descriptions of how to select distributions are in Chapter 2, *Understanding the Terminology*, and Chapter 3, *Setting Up and Running a Simulation*.

Defining Testing Costs: The Uniform Distribution

So far, Vision Research has spent \$10,000,000 developing ClearView and expects to spend an additional \$3,000,000 to \$5,000,000 to test it, based on the cost of previous tests. For this variable, "testing costs," Vision Research thinks that any value between \$3,000,000 and \$5,000,000 has an equal chance of being the actual cost of testing.

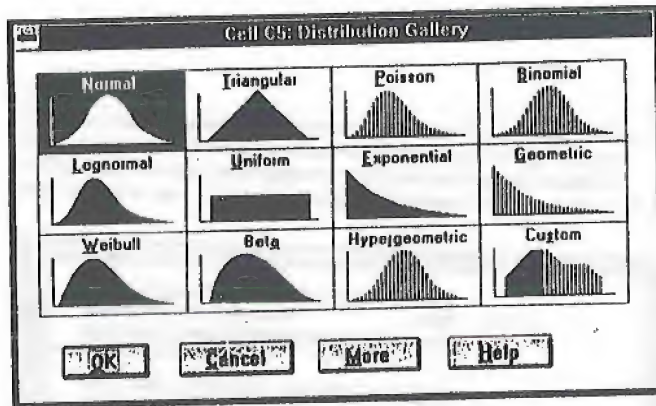
Using Crystal Ball, Vision Research chooses the Uniform distribution to describe the testing costs. The Uniform distribution describes a situation where all values between the minimum and maximum values are equally likely to occur, so this distribution best describes the cost of testing ClearView.

Once you choose the correct distribution type, you are ready to define the assumption cell.

To define the assumption cell for testing costs:

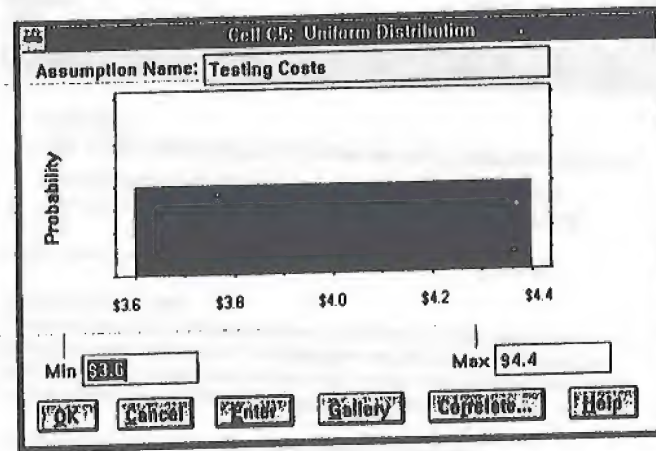
1. Click cell C5.
2. Choose Define Assumption from the Cell menu.

Crystal Ball displays a dialog box showing the Distribution Gallery.



3. Click the Uniform Distribution.
4. Click OK.

Crystal Ball displays a dialog box showing the Uniform distribution you chose for C5.



Since cell C5 already has a name next to it on the spreadsheet, the name is displayed in the dialog box. Use that name, rather than typing a new one. Also, notice that Crystal Ball assigns default values to the distribution. The method Crystal Ball uses to assign these

default values to each distribution is explained in Appendix E, Default Names and Distribution Parameters.

The Uniform distribution has two parameters—minimum and maximum. Vision Research expects to spend a minimum of \$3,000,000 and a maximum of \$5,000,000 on testing. Use these values in place of the defaults to specify the parameters of the Uniform distribution in Crystal Ball, as described in the following steps:

5. Type 3 in the Min box. (Remember, the numbers on the spreadsheet represent millions of dollars.)

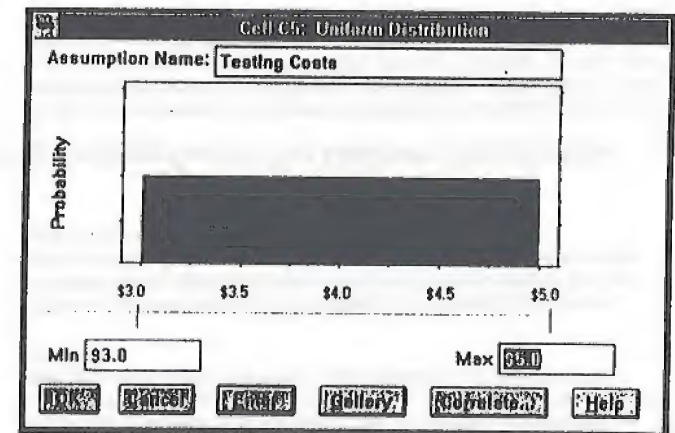
This represents \$3,000,000, the minimum amount Vision Research estimates for testing costs.

6. Press Tab and type 5 in the Max box.

This represents \$5,000,000, the maximum estimate for testing costs.

7. Click Enter.

The distribution changes to reflect the values you entered.



The distribution changes to reflect the values you entered

If you entered the values correctly, your screen looks like the example above. If you think you made a mistake, repeat steps 5 through 7. Later, when you run the simulation, Crystal Ball will generate random values for cell C5 that are evenly spread between 3 and 5 million dollars.

8. Click OK to return to the spreadsheet.

Defining Marketing Costs: The Triangular Distribution.

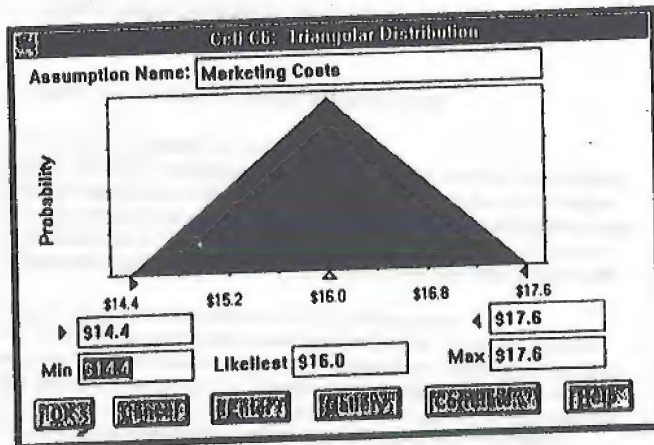
Vision Research plans to spend a sizeable amount marketing ClearView if it is approved by the FDA. They expect to hire a large sales force and kick off an extensive advertising campaign to educate the public about this exciting new product. Including sales commissions and advertising costs, Vision Research expects to spend between \$12,000,000 and \$18,000,000, most likely \$16,000,000.

Vision Research chooses the Triangular distribution to describe marketing costs because the Triangular distribution describes a situation where you can estimate the minimum, maximum, and most likely values to occur.

To define the assumption cell for marketing costs:

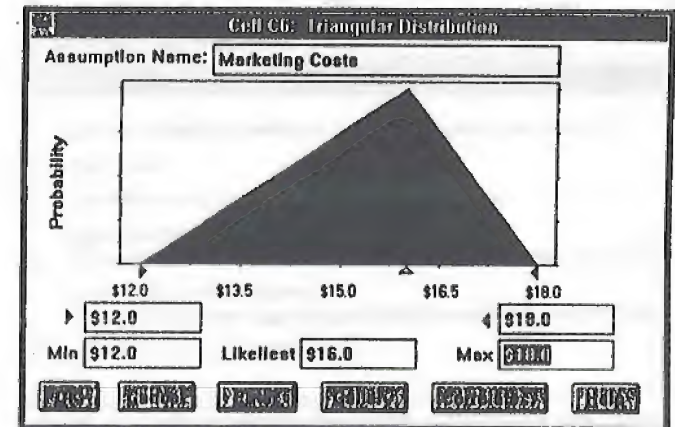
1. Click cell C6.
2. Choose Define Assumption from the Cell menu.
Crystal Ball displays the dialog box for the Distribution Gallery.
3. Click the Triangular Distribution.
4. Click OK.

Crystal Ball displays a dialog box showing the Triangular distribution you chose for C6.



Now specify the parameters for the Triangular distribution. As you can see in the example above, the parameters for the Triangular distribution are different from those specified earlier for the Uniform distribution. The Triangular distribution has three parameters—minimum, maximum, and likeliest. The following steps explain how to enter the parameters of the Triangular distribution:

5. Type 12 in the Min box.
This represents \$12,000,000, the minimum amount Vision Research estimates for marketing costs.
6. Press Tab to access the Likeliest box. If it does not contain the value 16, type 16.
This represents \$16,000,000, the most likely amount for marketing costs.
7. Press Tab and type 18 in the Max box.
This represents \$18,000,000, the maximum estimate for marketing costs.
8. Click Enter.
The distribution changes to reflect the values you entered.



When you run the simulation, Crystal Ball will generate random values that center around 16, with fewer values near 12 and 18.

9. Click OK to return to the spreadsheet.

Defining Patients Cured: The Binomial Distribution

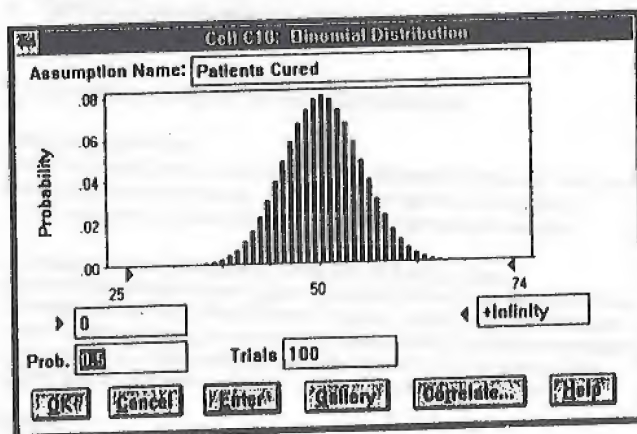
Before the FDA will approve ClearView, Vision Research must conduct a controlled test on a sample of 100 patients for one year. The FDA has stipulated that they will approve ClearView if it completely corrects the nearsightedness of 20 or more of these patients without any significant side-effects. In other words, 20% or more of the patients tested must show corrected vision after taking ClearView for one year. Vision Research is very encouraged by their preliminary testing, which shows a success rate of around 25%.

For this variable, "patients cured," Vision Research only knows that their preliminary testing shows a cure rate of 25%. Will ClearView meet FDA standards? Using Crystal Ball, Vision Research chooses the Binomial distribution to describe the uncertainties in this situation because the Binomial distribution describes the number of successes (25) in a fixed number of trials (100).

To define the assumption cell for patients cured:

1. Click cell C10.
2. Choose Define Assumption from the Cell menu.
Crystal Ball displays the Distribution Gallery dialog box.
3. Click the Binomial Distribution.
4. Click OK.

Crystal Ball displays the Binomial distribution (notice that the default value for the probability parameter is 0.5 or 50%).



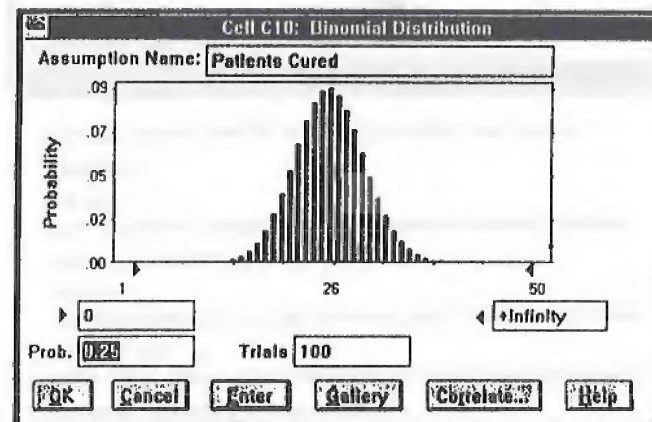
The Binomial distribution has two parameters—probability (prob) and trials. You know that Vision Research experienced a 25% success rate during preliminary testing, so use the value .25 for the probability parameter to show the likelihood or probability of success.

Crystal Ball Note: All probabilities can be expressed either as decimal fractions between 0 and 1, such as .03, or as whole numbers followed by the percent sign, such as 3%.

You also know the FDA expects Vision Research to test 100 people, so use the value 100 for the trials parameter. The following steps explain how to enter these parameters in the Binomial distribution.

5. Type .25 in the Prob box.
This represents the 25% likelihood or probability of successfully correcting nearsightedness.
6. The Trials box should contain the value 100. If it does not, press Tab and type 100 in the Trials box.
This represents the 100 patients in the FDA test.
7. Click Enter.

The distribution changes to reflect the values you entered:



When you run the simulation, Crystal Ball will generate random integer values between 0 and 100, simulating the number of patients that would be cured in the FDA test.

8. Click OK to return to the spreadsheet.

Defining Growth Rate: The Custom Distribution

Vision Research has determined that 40,000,000 people in the United States are currently afflicted with nearsightedness, and that an additional 0% to 5% will develop this condition during the year in which testing will take place.

However, the marketing department has learned that there is a 25% chance that a competing product will be released on the market soon. This product would decrease ClearView's potential market by 5% to 15%.

This variable, "growth rate of nearsightedness," cannot be described by any of the standard probability distributions. Since the uncertainties in this situation require a unique approach, Vision Research chooses Crystal Ball's Custom distribution to define growth rate. For the most part, the Custom distribution is used to describe situations that other distribution types cannot.

The method for specifying parameters in the Custom distribution is quite unlike the other distribution types, so follow the directions carefully. If you make a mistake, click Gallery to return to the Distribution Gallery, then start again at step 3.

Use the Custom distribution to plot both the potential increase and decrease of ClearView's market.

To define the assumption cell for the growth rate of nearsightedness:

1. Click cell C15.
2. Choose Define Assumption from the Cell menu.
Crystal Ball displays the Distribution Gallery dialog box.
3. Click the Custom Distribution.
4. Click OK.

Crystal Ball displays the Custom distribution dialog box (notice in the example on the next page that the chart area remains empty until you enter the values for the distribution).

To enter the first range of values:

5. Type 0% in the Value box.
This represents a 0% increase in the potential market.
6. Press Tab and type 5% in the Value2 box.
This represents a 5% increase in the potential market.

The box remains empty until you enter values.

7. Press Tab and type 75% in the Prob box.

This represents the 75% chance that Vision Research's competitor will not enter the market and reduce Vision Research's share.

8. Click Enter.

Crystal Ball displays a uniform distribution for the range 0.00% to 5.00%.

Uniform distribution for the first range of values.

To enter the second range of values:

9. Type -15% in the Value box.

This represents a 15% decrease in the potential market.

10. Press Tab and type -5% in the Value2 box.

This represents a 5% decrease in the potential market.

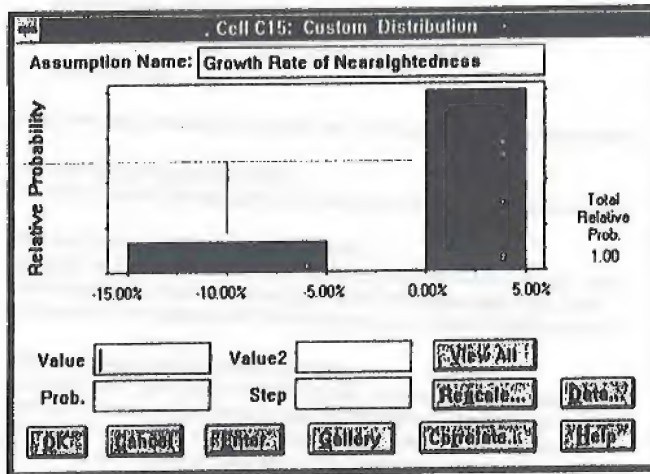
11. Press Tab and type 25% in the Prob box.

This represents the 25% chance that Vision Research's competitor will enter the market place and decrease Vision Research's share by 5% to 15%.

12. Click Enter.

Crystal Ball displays a Uniform distribution for the range -15% to -5%. Both ranges are now displayed in the custom distribution dialog box.

Crystal Ball displays both ranges.



In Chapter 2, *Understanding the Terminology*, you will learn to use a special feature on the Custom distribution dialog box—the Data button. You can use the Data button to pull numbers from specified cell ranges on the spreadsheet rather than typing them in the Custom distribution dialog box. When you run the simulation, Crystal Ball generates random values within the ranges you specified.

13. Click OK to return to the spreadsheet.

Defining Market Penetration: The Normal Distribution

Glossary Term:
Standard Deviation - The square root of the variance for a distribution. A measurement of the dispersion of values around the mean.

Glossary Term:
Mean or Mean Value - The familiar arithmetic average of a set of numerical observations (the sum of the observations divided by the number of observations).

Glossary Term:
Variance - The square of the standard deviation, i.e., the average of the squares of the deviations of a number of observations from their mean value.

The marketing department estimates that Vision Research's eventual share of the total market for the product will be normally distributed around a mean value of 8% with a standard deviation of 2%. "Normally distributed" means that Vision Research expects to see the familiar bell-shaped curve with about 68% of all possible values for market penetration falling between one standard deviation below the mean value and one standard deviation above the mean value, or between 6% and 10%. The low mean value of 8% is a conservative estimate that takes into account the side-effects of the drug that were noted during preliminary testing. In addition, the marketing department estimates a minimum market of 5%, given the interest shown in the product during preliminary testing.

Vision Research chooses the Normal distribution to describe the variable "market penetration."

To define the assumption cell for market penetration:

1. Click cell C19.

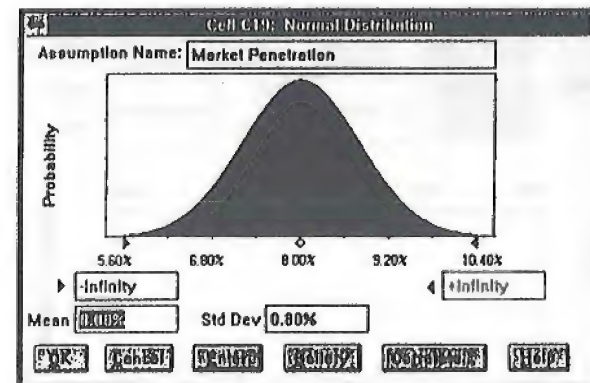
2. Choose Define Assumption from the Cell menu.

Crystal Ball displays the Distribution Gallery dialog box.

3. Click the Normal Distribution.

4. Click OK.

Crystal Ball displays a dialog box showing the Normal distribution you chose for cell C19.



Now specify the parameters for the Normal distribution: the mean and the standard deviation.

5. The Mean box should contain the value 8.00%. If it does not, type 8% in the Mean box.
This represents an estimated average for market penetration of 8%.

6. Press Tab and type 2% in the Std Dev box.

This represents an estimated 2% standard deviation from the mean.

7. Click Enter.

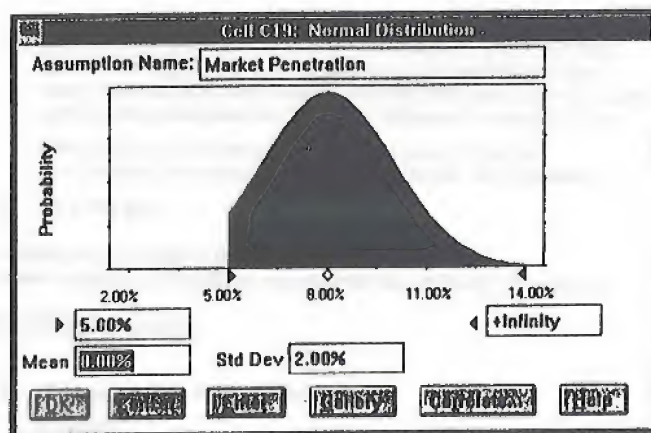
Crystal Ball scales the Normal distribution to fit the chart area so the shape of the distribution does not change. However, the percent range at the bottom of the chart does change.

8. Press Tab twice and type 5% in the left end-point grabber box.

This represents 5%, the minimum market for the product.

9. Click Enter.

The distribution changes to reflect the values you entered.



When you run the simulation, Crystal Ball will generate random values that follow a Normal distribution around the mean value of 8%, and no values will be generated below the 5% minimum limit.

10. Click OK to return to the spreadsheet.

Defining Forecasts

Now that you have defined the assumption cells in your model, you are ready to define the forecast cells. The forecast cells contain the formulas that refer to one or more assumption cells.

The president of Vision Research would like to know both the likelihood of achieving a profit on the product and the most likely profit, regardless of cost. Therefore, the president is interested in both gross profit (cell C21) and net profit (cell C23) for the ClearView project.

Calculating Gross Profit

Crystal Ball can generate more than one forecast when running a simulation. In this case, you will want to define both the gross profit and net profit formulas as forecast cells. First, look at the contents of the cell for gross profit:

1. Click cell C21.

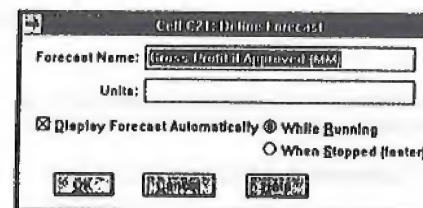
The cell contents are displayed in the formula bar near the top of your screen. The contents are $C16 \cdot C19 \cdot C20$. Crystal Ball will use this formula to calculate gross profit by multiplying Persons With Nearsightedness After One Year (C16) by Market Penetration (C19) by Profit Per Customer (C20).

Now that you understand the gross profit formula, you are ready to define the forecast formula cell for gross profit.

To define the forecast cell for gross profit:

2. Choose Define Forecast from the Cell menu.

The Define Forecast dialog box is displayed. You may enter a name for the forecast. Since the forecast cell has a name next to it on the spreadsheet, that name is displayed in the dialog box.



Glossary Term:
Forecast Formula -
A formula that has
been defined as a
forecast cell.

Use the forecast name that is displayed, rather than typing a new name.

Next, you will indicate that the forecast chart is in millions of dollars, since the spreadsheet model involves millions of dollars, and request that the forecast chart be displayed during the simulation:

3. Press Tab and type millions in the Units box.
4. Click the Display Forecast Automatically box, if it is not already checked.
5. Click OK to return to the spreadsheet.

Calculating Net Profit

Before defining the forecast cell formula for net profit, look at the contents of the cell for net profit:

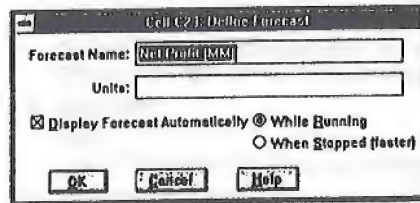
1. Click cell C23.

The contents are displayed in the formula bar near the top of your screen. The contents are: IF(C11,C21-C7,-C4-C5).

The formula translates as: If the FDA approves the drug (C11 is true), then calculate net profit by subtracting total costs (C7) from gross profit (C21). However, if the FDA does not approve the drug, (C11 is false), then calculate net profit by deducting both development costs (C4) and testing costs (C5) incurred to date.

To define the forecast cell for net profit:

2. Choose Define Forecast from the Cell menu.
- A dialog box is displayed.



Again, use the forecast name that is displayed in the Forecast Name box, specify millions in the Units box, and check the Display Forecast Automatically box.

3. Press Tab and type millions in the Units box.
4. Click the Display Forecast Automatically box, if it is not already checked.
5. Click OK to return to the spreadsheet.

You have defined assumptions and forecast cells for the "Vision Research" spreadsheet, now you are ready to run a simulation.

Running a Simulation

When you run a simulation in Crystal Ball, you have the freedom to stop and then continue the simulation at any time. The Run, Stop, and Continue commands appear on the Run menu as you need them. For example, while you are running a simulation, the Stop command appears at the top of the menu. If you stop the simulation, the Continue command takes its place. Practice using these commands when you run the simulation for the ClearView project.

Before you begin the simulation, specify the number of trials and Initial seed value so your simulation will look like the forecast charts in this tutorial. In Chapter 4, Interpreting the Results, trials and seed value are discussed in detail.

Glossary Term:
Seed Value - The first number in a sequence of random numbers. A given seed value will produce the same sequence of random numbers every time you run a simulation.

To specify the number of trials and Initial seed value:

1. Choose Run Preferences from the Run menu.
Crystal Ball displays the Run Preferences dialog box.
2. Type 200 in the Maximum Number of Trials box.
3. Click the Use Same Sequence of Random Numbers and type 1000 in the Initial Seed Value box.
The value 1000 is used as an arbitrary number for the Initial seed value.
4. Click OK.

Now practice using the Run, Stop, and Continue commands:

1. Choose Run from the Run menu.
Crystal Ball displays the net profit forecast chart neatly stacked on top of the Gross Profit forecast chart. As the simulation proceeds, the forecast charts reflect the changing values in the forecast cells.
2. Choose Stop from the Run menu on the front forecast window.

Crystal Ball updates the forecast charts to reflect the current values in the forecast cells. You can also stop the simulation by pressing Alt-U, O.

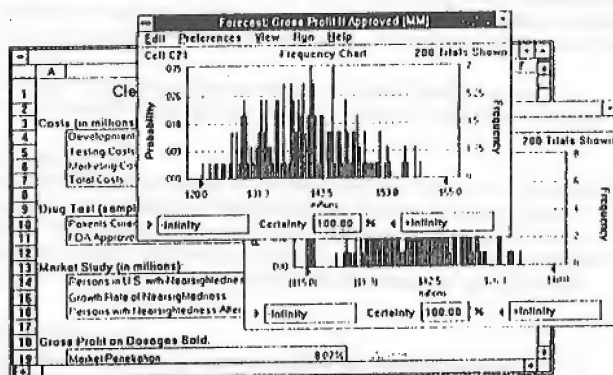
3. Choose Continue from the Run menu on the front forecast window.

Crystal Ball continues the simulation. You can also continue the simulation by pressing Alt-U, U.

You may not be able to see two complete forecast charts at the same time. However, there are several ways to bring individual forecast windows to the front of the window stack. The easiest way is to click on the forecast window if it is visible.

1. Click on the Gross Profit If Approved forecast window.

The Gross Profit forecast chart is displayed.

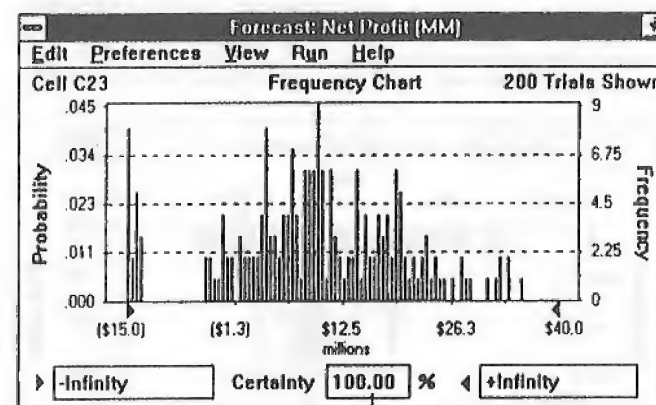


2. Choose Forecast Windows from the Run Menu on the front forecast window. Click Open All Forecasts to move the Net Profit forecast chart to the front again.

Excel Note: In Excel, each forecast window has its own menu bar. In Lotus 1-2-3, there is one menu bar for all the forecast windows.

While the simulation is running, Crystal Ball displays a frequency distribution for each forecast to reflect the changing values in the forecast cell. The frequency distribution is displayed as columns on your screen.

3. Continue to run the simulation until it stops at 200 trials.



A frequency distribution shows the number or frequency of values occurring in a given group interval. In the example above, the frequency distribution on the Net Profit forecast chart shows a frequency of 9 for the group interval that contains the most values. That means 9 values occurred in the group interval. Chapter 4, *Interpreting the Results*, describes how the forecast report provides a list of the group intervals.

Chapter 3, *Setting Up and Running a Simulation*, and Chapter 4, *Interpreting the Results*, describe the forecast chart in more detail. For now, remember that the Forecast chart graphs the forecast results and shows how the forecast values are distributed. As the simulation progresses, Crystal Ball continues to update the frequency distribution for each forecast cell and the forecast results become more accurate.

Interpreting the Results

Now that you have run the simulation, you are ready to interpret the forecast results in more depth. The president of Vision Research faces a difficult decision—should the company scrap the ClearView project or proceed to develop and market this revolutionary new drug? To examine this question you need to look at the forecast chart in more detail.

Understanding the Forecast Chart

Excel Note: Crystal Ball windows are separate from Excel windows. If Crystal Ball's windows disappear from your screen, they are usually simply behind the main Excel window and you can bring them to the front by pressing Alt-Tab.

Crystal Ball forecasts the entire range of results for the Vision Research project. However, the forecast charts show only the display range which by default is set to exclude the most extreme values in the forecast. Chapter 4, *Interpreting the Results*, describes how to change the display range to examine specific sections of the forecast in greater detail.

In this example, the display range includes the values from minus \$15.00 to \$40.00, as shown on the Net Profit forecast chart.

The forecast chart also shows the certainty range for the forecast. By default, the certainty range includes all values from negative infinity to positive infinity. In the next section, you will learn to change the certainty range.

Crystal Ball compares the number of values in the certainty range with the number of values in the entire range to calculate the certainty level. The example above shows a certainty level of 100%, since the initial certainty range includes all possible values.

Remember, the certainty level is an approximation, since the spreadsheet model can only approximate the elements of the real world.

In the upper-right corner of the forecast chart, Crystal Ball shows the number of trials. This indicates the number of trials currently being shown in the display range. Since the display range is initially set by default to exclude extreme values from being displayed, this number may sometimes be less than the total number of trials.

Determining the Certainty Level

Now the Vision Research president wants to know how certain Vision Research can be of achieving a profit and what are the chances of a loss.

To determine the certainty level of a specific value range:

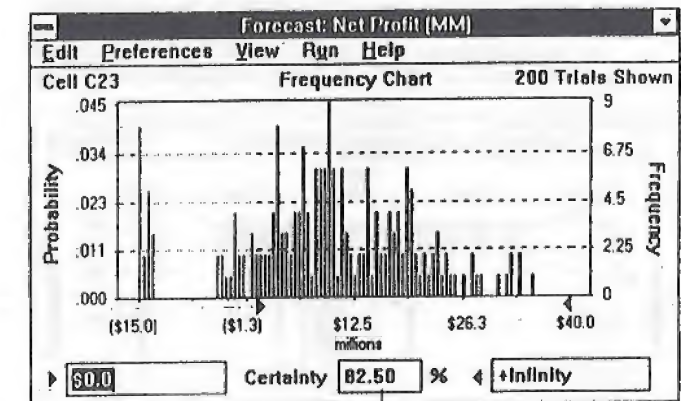
1. Press Tab twice and type 0 in the range minimum box on the Net Profit forecast chart.
2. Press Enter.

Crystal Ball moves the left end-point grabber to the break-even value of \$0.0 and recalculates the certainty level.

Left end-point
grabber

Range minimum box

Certainty Level

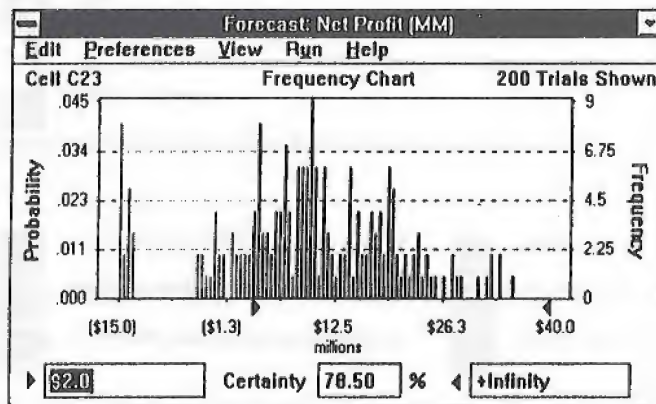


Analyzing the Net Profit forecast chart again, you can see that the value range between the end-point grabbers shows a certainty level of 82.5%. That means that Vision Research can be 82.5% certain of achieving a net profit. You can therefore calculate an 17.5% chance of suffering a net loss (100% minus 82.5%).

Now the president of Vision Research would like to know the certainty of achieving a minimum profit of \$2,000,000. With Crystal Ball you can easily answer this question.

3. Type 2 in the range minimum box.
4. Press Enter.

Crystal Ball moves the left end-point grabber to \$2.0 and recalculates the certainty level.

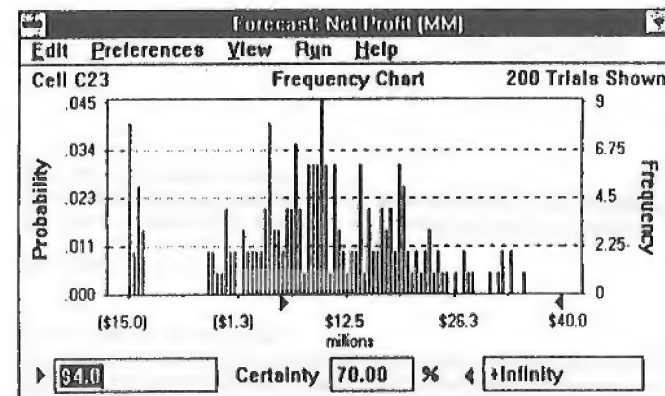


Vision Research can be 78.5% certain of achieving a minimum net profit of \$2,000,000.

Vision Research is very encouraged by the forecast results. The president now wants to know how certain Vision Research can be of achieving a minimum net profit of \$4,000,000. If Crystal Ball shows that Vision Research can be at least two-thirds certain of a \$4,000,000 net profit, the president is ready to go ahead with the ClearView project. Again, Crystal Ball can easily answer this question.

5. Type 4 in the range minimum box.
6. Press Enter.

Crystal Ball moves the left end-point grabber to \$4.0 and recalculates the certainty level.



The Net Profit forecast chart (above) shows a certainty level of 70%. With 70% certainty of a minimum net profit of \$4,000,000, Vision Research decides to go ahead with the ClearView project and proceed to develop and market this revolutionary new drug.

The president of Vision Research also is interested in the most likely profit regardless of cost. You now can analyze the gross profit forecast chart as you did the net profit chart.

Summary

In this tutorial, you have explored only a few questions that Vision Research might ask as they analyze the results of the simulation. As you read through this manual, you will learn to explore the forecast results in more depth. For example, you can customize the forecast charts, create trend charts, analyze the sensitivity of the model, interpret the descriptive statistics, and print comprehensive reports for any simulation. Crystal Ball provides all these features so that you can be confident about achieving the results you are looking for.

The Crystal Ball Toolbar

To aid in setting up spreadsheet models and running simulations, Version 3.0 of Crystal Ball comes with a customized toolbar that provides instant access to the most commonly used menu commands.

The Excel toolbar looks like this:



The buttons in the first three groups are from the Cell menu:



Define Assumption



Define Forecast



Select All Assumptions



Select All Forecasts



Copy Assumptions/Forecasts (Excel)



(Lotus 1-2-3)



Paste Assumptions/Forecasts (Excel)



(Lotus 1-2-3)



Clear Assumptions/Forecasts (Excel)



(Lotus 1-2-3)

The buttons from the next four groups are from the Run menu:



Run Preferences



Run



Stop



Reset



Single Step



Forecast Windows



Open Trend Chart



Open Sensitivity Chart



Create Report



Extract Data

The last button is from the Excel Help menu:



Help (Excel)



(Lotus 1-2-3)



Open Crystal Ball

(Lotus 1-2-3)

In Excel...

When you close Crystal Ball, it remembers the state of the toolbar. If the toolbar was showing when you closed Crystal Ball, it will be automatically revealed the next time you open Crystal Ball. If the toolbar was hidden when you closed Crystal Ball, it will remain hidden the next time you start Crystal Ball.

If you don't want to use the Crystal Ball toolbar, select the Toolbars command from the Options menu. The Toolbars dialog box will appear. Select Crystal Ball from the Show Toolbars list (it's probably hiding at the end of the list; click on the bottom scroll arrow to reveal it). Click the Hide button. This hides the toolbar. You can get it back by clicking the Show button.

In Lotus 1-2-3...

The Lotus toolbar works and looks very much like the Excel toolbar. To show and hide the toolbar, use the pop-up menu on the bottom of the Lotus 1-2-3 main window.

Closing Crystal Ball

At this point, you can close Crystal Ball and continue reading to learn how Crystal Ball takes the risk out of your spreadsheet analysis.

To close Crystal Ball

1. Choose **Close Crystal Ball** from the **Run** menu on the menu bar.

A dialog box is displayed, asking you to confirm your decision. If you click **OK**, the **Cell** and **Run** menus are removed from the menu bar and Crystal Ball is closed. However, the "Vision Research" spreadsheet will remain on your screen.

Glossary Term:
Forecast Definition -
The forecast name and
parameters assigned to
a cell in a Crystal Ball
dialog box.

Glossary Term:
Forecast Value -
A value calculated by the
forecast formula during
an iteration. These values
are kept in a list for each
forecast, and are
summarized graphically
in the forecast chart and
numerically in the
descriptive statistics.

Crystal Ball Note: *Crystal Ball will also close automatically when you exit from the spreadsheet application.*

Crystal Ball keeps your assumption and forecast definitions (but not the forecast values) with the spreadsheet. When you save your spreadsheet, the definitions are saved with it. To learn about saving forecast values, see the **Save Run/Restore Run** section in Chapter 3, *Setting Up and Running a Simulation*.

Understanding the Sensitivity Chart

The Sensitivity Chart feature provides you with the ability to quickly and easily judge the influence each assumption cell has on a particular forecast cell. During a simulation, Crystal Ball ranks the assumptions according to their importance to each forecast cell. The Sensitivity Chart displays these rankings as a bar chart, indicating which assumptions are the most important or least important ones in the model. You can output (print) the Sensitivity Chart on the report or copy it to the clipboard.

The Sensitivity Chart feature provides three key benefits:

1. You can find out which assumptions are influencing your forecasts the most, reducing the amount of time needed to refine estimates.
2. You can find out which assumptions are influencing your forecasts the least, so that they may be ignored or discarded altogether.
3. As a result, you can construct more realistic spreadsheet models and greatly increase the accuracy of your results because you will know how all of your assumptions affect your model.

Creating the Sensitivity Chart

In the examples directory on your Crystal Ball disk there is a "Toxic Waste Site" spreadsheet you can use to experiment with the Sensitivity Chart feature. The Sensitivity Chart you create will display, in descending order, the assumptions in a risk assessment of a toxic waste site. The assumption with the highest level of sensitivity can be considered as the most influential assumption in the model.

To create a Sensitivity Chart:

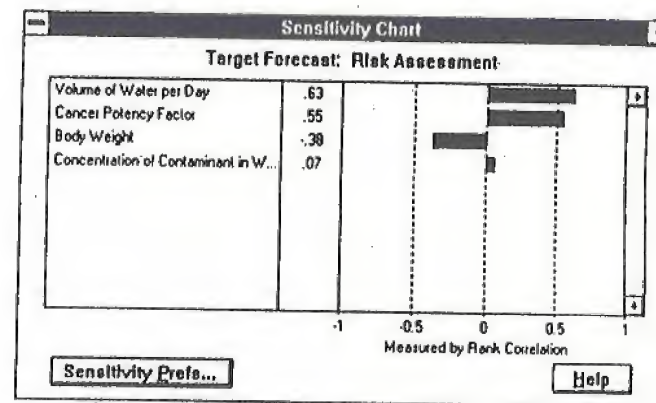
1. Close any spreadsheet windows that are currently open.
2. Choose Open from the File menu.
3. Open the TOXIC.XLS (Excel) or TOXIC.WK4 (Lotus) spreadsheet.
4. Choose the Sensitivity Analysis option in the Run Preferences dialog box.

Run Options

☒ Sensitivity Analysis ☐ Correlations Off

5. Run a simulation
6. Stop the simulation.
7. Select Open Sensitivity Chart from the Run menu.

A window will open displaying the sensitivity rankings of the assumptions in your simulation.



If you select Open Sensitivity Chart but forgot to make the appropriate selection in the Run Preferences dialog box, you will need to reset the simulation and run it again.

The assumptions (and possibly other forecasts) are listed on the left side, starting with the assumption with the highest sensitivity. Assumptions appear as green bars and forecasts appear as blue bars unless you change the color settings. Use the scroll bar to view the entire bar chart.

In this example, there are four assumptions listed in the Sensitivity Chart. The first assumption, Volume of Water per Day, has the highest sensitivity ranking and can be considered the most important assumption in the model. A researcher running this model would want to investigate this assumption further in the hopes of reducing its uncertainty, and therefore its effect on the target forecast. The last assumption, Concentration of Contaminant in Water, has the lowest sensitivity ranking and is the least important assumption in the model. The effect of this assumption on the target forecast is not as great as the others and, in this particular case, could be ignored or eliminated as an assumption altogether. Sensitivity charts like this one illustrate that one or two assumptions typically have a dominant effect on the uncertainty of a forecast.

How Crystal Ball Calculates Sensitivity

Crystal Ball calculates sensitivity by computing Spearman rank correlation coefficients between every assumption and every forecast cell while the simulation is running. Correlation coefficients provide a meaningful measure of the degree to which assumptions and forecasts *change together*. If an assumption and a forecast have a high correlation coefficient, it means that the assumption has a significant impact on the forecast (both through its uncertainty and its model sensitivity). Positive coefficients indicate that an increase in the assumption is associated with an increase in the forecast. Negative coefficients imply the reverse situation. The larger the absolute value of the correlation coefficient, the stronger the relationship.

Crystal Ball also computes the correlation coefficients for all pairs of forecasts while the simulation is running. You may find this sensitivity information useful if your model contains several intermediate forecasts that feed into a final forecast.

An option in the Sensitivity Preference dialog box lets you display the sensitivities as a percentage of the contribution to the variance of the target forecast. This option, called Contribution to Variance, doesn't change the order of the items listed in the Sensitivity Chart and makes it easier to answer questions such as "what percentage of the variance or uncertainty in the target forecast is due to assumption X?". However, it is important to note that this method is only an approximation and is *not precisely* a variance decomposition. Crystal Ball calculates Contribution to Variance by squaring the rank correlation coefficients and normalizing them to 100%.

Caveats

The Sensitivity Chart feature has several limitations you should be aware of:

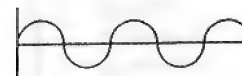
1. The sensitivity calculation may be inaccurate for correlated assumptions. For example, if an important assumption were highly correlated with an unimportant one, the unimportant assumption would likely have a high sensitivity with respect to the target forecast. Assumptions that are correlated are flagged as such on the Sensitivity Chart. In some circumstances, turning off correlations in the Run Preference dialog box may help you to gain more accurate sensitivity information.

2. The sensitivity calculation may be inaccurate for assumptions whose relationships with the target forecast are not monotonic. A monotonic relationship means that an increase in the assumption tends to be accompanied by a strict increase in the forecast; or an increase in the assumption tends to be accompanied by a strict decrease in the forecast.

For example, the relationship $y = \text{Log}(x)$ is monotonic:



While $y = \text{Sin}(x)$ is not:



Customizing the Sensitivity Chart

Use the Sensitivity Prefs dialog box to customize the Sensitivity Chart. As you become more familiar with the Sensitivity Chart, practice selecting preferences that help you get the answers you are looking for and are appropriate for the data you are working with.

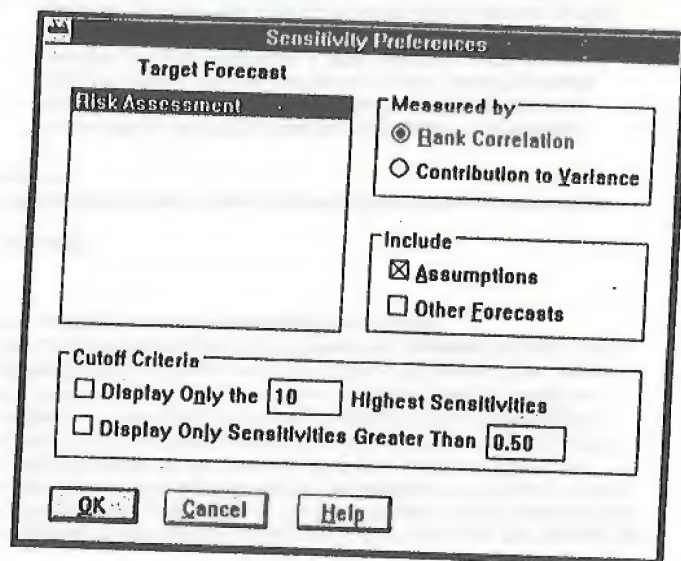
1. Click Sensitivity Prefs to open the Sensitivity Preferences dialog box (an example appears at the top of the following page).

The Target Forecast list box allows you to choose which forecast cell is the target of the sensitivity analysis.

The Measure by option allows you to determine if the bar chart will show the sensitivities as rank correlations or contributions to variance. Rank correlations range from -1 to +1 and indicate both magnitude and direction. Contributions to variance range from 0% to 100% and indicate relative importance.

The Include option lets you select which types of Crystal Ball data to be ranked against the target forecast. You have three display options: Assumptions, Forecasts, or Both (by selecting both Assumptions and Forecasts).

Chapter 4 Interpreting the Results



Crystal Ball will always include *all* the Assumptions and Forecasts in your model even though they may be unrelated. Generally, this is not an issue since unrelated assumptions and forecasts will have sensitivity rankings close to zero. However, correlation may affect sensitivity analysis if there is a strong correlation between the variables and at least one of the variables is highly sensitive.

The Cutoff Criteria option gives you an even greater level of control over how many sensitivities appear on the Sensitivity Chart list by allowing you to assign values for count cutoff, value cutoff, or both.

Creating Reports

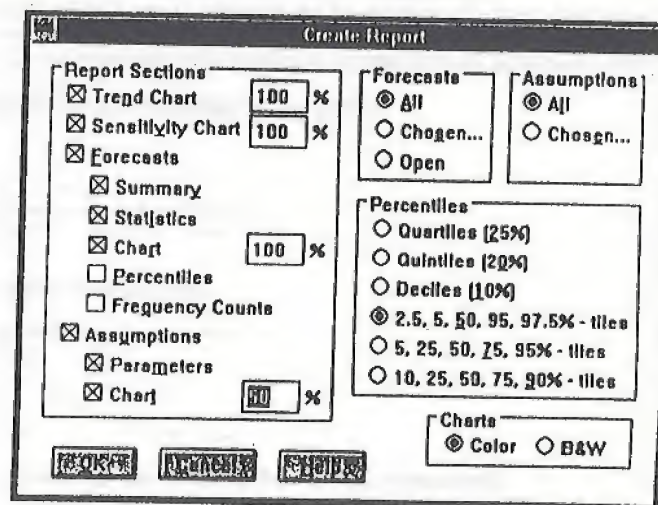
A report can be created for each forecast using the Report command. Any or all of the following items can be included in the report using the Report dialog box:

- Trend charts
- Sensitivity charts
- Forecast summaries

- Statistics
- Forecast charts
- Percentiles
- Frequency counts
- Assumption parameters
- Assumption charts

To create a report Choose *Create Report* from the Run menu.

The Create Report dialog box is displayed.



Select information to be included in the report with the following steps:

1. To include all forecasts in the report, click *All* in the Forecasts options.
2. To include only selected forecasts, and to specify the order in which they will appear in the report, click *Chosen* in the Forecasts options.

A dialog box is displayed allowing you to choose from a list of available forecasts.

3. To include only those forecasts which are currently open, click Open in the Forecasts options.
4. To include all the assumption information, click All in the Assumptions options.
5. To include only selected assumptions, and to specify the order in which they will appear in the report, click Chosen in the Assumptions options.
A dialog box is displayed allowing you to choose from a list of available forecasts.
6. To include the trend chart in the report, click the Trend Chart check box in the Report Sections options. To reduce or enlarge the size of the trend chart, type the percentage in the % box.
7. To include the sensitivity chart in the report, click the Sensitivity Chart check box in the Report Sections options. To reduce or enlarge the size of the sensitivity chart, type the percentage in the % box.
8. To include forecast information, click the Forecasts check box in the Report Sections options.
The Forecasts option can include the following subsections:
 9. To include a brief summary of the forecast results, click the Summary check box.
 10. To include the statistics, click the Statistics check box.
 11. To include the forecast chart, click the Chart check box. Type the percentage in the % box to reduce or enlarge the size of the forecast chart.
 12. To include a list of percentiles, click the Percentiles check box.
The Percentiles option shows the certainty of achieving a particular value level. For example, if a report was printed for the Net Profit forecast discussed in the "Vision Research" tutorial in Chapter 1, the Percentiles section might show that you could be 75% certain of achieving a net profit or loss below the threshold value of \$17 million.
 13. To include a list of frequency counts and the cumulative counts for the forecast's group intervals, click the Frequency Counts check box.
The Group Intervals shows the starting value, ending value, probability, and frequency for each group interval.
 14. To include assumption information, click the Assumptions check box in the Report Sections options.
The Assumptions option can include the following subsections:

15. To include the assumption parameter information, click the Parameters check box.
16. To include the assumption chart, click the Chart check box. To reduce or enlarge the size of the assumption chart, enter the percentage in the % box.

If you choose Percentiles from the Report Sections options, you now choose the percentiles you want displayed.

17. To select which percentiles to include in the report, click the appropriate button in the Percentiles options.
The Percentiles options for the forecast report show the certainty of achieving a value below a particular threshold. The first option divides the frequency distribution into quartiles (four sections), showing the value levels for the following percentiles: 0%, 25%, 50%, 75%, and 100%. The next option divides the distribution into quintiles (five sections). The Deciles option creates 10 sections. The next three options show the value levels for the following levels, respectively:
 - 2.5%, 5%, 50%, 95%, and 97.5%
 - 5%, 25%, 50%, 75%, and 95%
 - 10%, 25%, 50%, 75%, and 90%
18. Click OK.

Crystal Ball creates the report as an Excel or Lotus 1-2-3 spreadsheet. You can modify, print, or save the report in the same way as any other spreadsheet.

Crystal Ball Note: To suppress the initial report header and any cell references that occur in the report, hold down the shift key when choosing Report from the Run menu.

Crystal Ball Note: If the simulation has not been run or if it has been reset, the report will include only an assumptions section. Options in the Report Selection dialog box affecting items other than assumptions will be disabled.

Braincel™

Version 2.0

Promised Land Technologies, Inc.
New Haven, CT
© All Rights Reserved, 1990-93.

WHARTON REPROGRAPHICS

12

Tutorial—A Sample Application With Braincel

The tutorial shows you how you can set up a sample application with Braincel. We'll show you how to teach a Braincel to make loan repayment probability forecasts based on past applications.

Important: The Tutorial application is simplified for clarity. A real-world Loan Expert would need a larger database than the one we are showing here.

Setting up a problem for Braincel

Gather data relevant to the problem you're presenting to the program.

To create our Braincel Loan Expert, we first decided exactly what we were looking to predict. We decided on Loan Repayment Ability. We will teach the Braincel Expert to determine how able each applicant is to repay a \$2000.00 personal loan.

Basic data was collected. For example, information on monthly income and expenses, how long the applicants had worked at their jobs, etc. Also collected was a human loan officer's decision on the each application, assessing the ability of the applicant to repay the loan on a scale of 1 to 5. A "1" means very poor loan repayment probability and "5" means excellent loan repayment probability.

Organize the data into two sets: inputs and outputs. Put the inputs and outputs in individual columns.

- An input is any data that is used by the expert to arrive at a solution, prediction or decision.
—For the Loan Expert, the inputs are the 8 pieces of information collected on each applicant.
- An output is the solution, prediction or decision that Braincel will be learning to produce.
—For the Loan Expert, the output is the loan officer's decision on the repayment ability of each applic.

Figure 1.1 BCDATA.XLS

✚ To see the data loaded for the Tutorial Loan Expert:
Open BCDATA.XLS from the Braincel directory.

This worksheet contains the data from 17 past loan applicants. Each applicant's information is in a separate row in the worksheet. The data is arranged database fashion, with each column as a separate input or output. We placed the output in the right-most column to make it easier to keep track of.

Allow for minimum and maximum values for each data column.

Minimum and maximum values let Braincel know what it can expect to see in each column. The program mathematically scales each column when it performs its calculations; the min/max values serve as the endpoints for this internal scaling procedure.

Braincel will automatically calculate minimum and maximum values for you. It uses two blank rows directly below the past loan applicants.

Note: Non-numerical data is treated separately. Refer to Variations—Using text in your data.

Define three data sets for training and testing

Your Braincel Expert needs to see the data in three different sets as part of its learning process.

Using the Excel *Define Name* command, define three data sets as named ranges within the total available data.

These ranges are called: the Training Range, the Test Range and the Predict Test Range. Naming these ranges with Excel makes it easier to reference them.

Define the Training Range.

This range should include approximately 90 % of your data records. This is the range that your Braincel Expert will learn from. Be sure to include two empty rows for minimum and maximum values

Figure 1.3 The Training Range highlighted
The highlighted cells represent the training range for this Expert. Include all input and output columns for each record, with the min and max rows.

✚ To define a Training Range on BCDATA.XLS:

1. Activate BCDATA.XLS, if necessary.
2. Select cells R7C3:R23C11. (Loan records 3–17 with the min and max rows)
3. Select **Formula Define Name**.
4. Name this range TRAINING_RANGE.

Define the Test Range.

Instead of including the output column when using the Excel **Define Name** command, include an empty column. Braincel uses this column to write the output it calculates for each record. For the Loan Expert, we have titled the empty column Calculated Output.

The Test Range is used to determine how well the Braincel Expert has learned to mimic the outputs it has seen during training.

	1	2	3	4	5	6	7	8	9	10	11	12
1												
2	Loan											
3	Record	Monthly	Monthly	Home	Yrs at	Yrs at	Yrs at	Yrs at	No. of			Calculated
4	Number	Income	Expenses	Owner?	Job	Job	Address	Address	Depend.	Output		Output
5	1	3000	2800	0	1	2	3	4	3	1		
6	2	4500	1500	1	6	4	4	9	3	4		
7	3	3000	1500	0	2	8	6	2	5	3		
8	4	850	425	1	3	3	25	25	1	3		
9	6	1000	3000	0	0.1	0.3	0.1	0.3	4	1		
10	6	9000	2250	1	3	4	5	3	2	6		
11	7	4000	1000	1	3	5	3	2	1	4		
12	8	3500	2500	0	0.5	0.5	0.5	2	1	1		
13	9	2300	1200	1	5	3	1	4	1	3		
14	10	4500	3500	0	3	3	10	1	5	2		
15	11	1200	1000	0	0.5	0.9	1	0.6	3	1		
16	12	800	800	0	0.1	1	9	1	3	1		
17	13	7500	3000	1	10	3	10	3	4	5		
18	14	3000	1000	1	20	8	15	10	1	6		
19	15	2500	700	1	10	9	15	9	3	9		
20	16	3000	2600	1	8	11	3	4	2	7		

Figure 1.4 The Test Range highlighted

The highlighted cells represent the Test Range. The Braincel Expert will write its calculations into the empty column later, during testing. You compare its calculation with the correct answer in the adjacent column. We are testing on a small subset of the Training Range, records 6–10.

✚ To define a Test Range on BCDATA.XLS:

1. Activate BCDATA.XLS, if necessary.
2. Select cells R10C3:R14C10, R10C12:R14C12 as a complex range. (Both blocks of cells should be highlighted at the same time.)
3. Select **Formula Define Name**.
4. Name this range TEST_RANGE.

Define the Predict Test Range.

This range consists of the 10% of your data that you withheld from the Training Range. The records were not in the Training or Test Ranges, so they'll be fresh data for the Expert after it's been trained, later in the Tutorial.

This range tests the predicting ability of your new Expert. By comparing the Expert's calculation against the historical output, you'll preview how accurate the Expert will be when given new data.

	1	2	3	4	5	6	7	8	9	10	11	12
1												
2	Loan											
3	Record	Monthly	Monthly	Home	Yrs at	Yrs at	Yrs at	Yrs at	No. of			Calculated
4	Number	Income	Expenses	Owner?	Job	Job	Address	Address	Depend.	Output		Output
5	1	3000	2800	0	1	2	3	4	3	1		
6	2	4500	1500	1	6	4	4	9	3	4		
7	3	3000	1500	0	2	8	6	2	5	3		

Figure 1.5 The Predict Test Range highlighted

✚ To define a Predict Test Range on BCDATA.XLS:

1. Activate BCDATA.XLS, if necessary.
2. Select cells R5C3:R6C10, R5C12:R6C12 as a complex range. (Both blocks of cells should be highlighted at the same time.)
3. Select **Formula Define Name**.
4. Name this range PREDICT_TEST.

Note: You may choose to continue the tutorial with BCDATA.XLS or you may open a worksheet that we have prepared, BCTUTOR.XLS.

Create an Expert file

Now that you have finished setting up the worksheet in Excel, you are ready to use Braincel.

✚ To create a Braincel Expert:

1. Select **Braincel Braincel Menu** in the Excel menu. The Braincel menu replaces the regular Excel menu.
2. Select **File New Expert**.
3. Fill the box as shown in Figure 1.6 below.

Figure 1.6 *New Expert* box properly filled in for the tutorial

Field Definitions

Network Name

Enter the file name of your expert — Tutorial. (You can use up to eight characters. Braincel assigns all files created from the *New Expert* dialog box the file extension .NET.)

Number of Inputs

Enter the number of input columns in the training range — 8.

Number of Outputs

Enter the number of output columns in the training range — 1.

Note: The Password is optional and is not used in the Tutorial. See Full Menu Reference—New Expert for more information.

Training the Expert

✚ To Begin Network Training:

1. Select **Expert Train Expert**
2. Select BCDATA.XLS or BCTUTOR.XLS, the worksheet where our training data is stored.
Note: Only worksheets open in Excel will appear in this box.
3. Select TRAINING_RANGE from the Ranges box
Note: All defined ranges on the selected worksheet will appear in the Ranges box
4. Add TRAINING_RANGE to Selected Ranges. Click OK.
5. Set error for 5% and training time for 10 minutes.
6. Allow Braincel to fill min/max rows. Press OK
7. Accept bounded output type. Press OK

Field Definitions

Stop At Error (%)

Error indicates how accurate the Expert is in calculating the output in your training data. Error refers to the average difference between an output and the corresponding Expert calculated output, scaled for the range of the output. (For more details, see page 38.) The Expert will stop training when it has reached the error that you specify.

Time (hh:mm:ss)

Time refers to how long you would like to train the network. The network will continue training until the time is elapsed or until the desired error is achieved, *whichever comes first*. Time is measured in hours, minutes, and seconds (hh:mm:ss).

Note on training time: Braincel's training speed depends on how powerful a computer you're using to train. A math coprocessor, RAM memory and CPU speed all determine how fast Braincel runs.

As soon as you press OK...
THE EXPERT IS NOW TRAINING

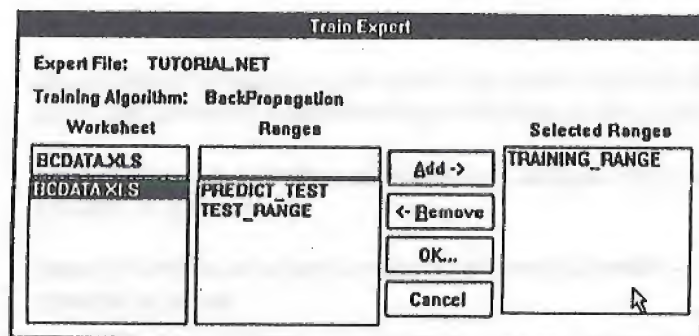
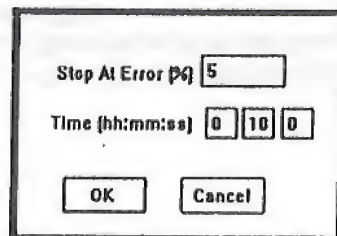
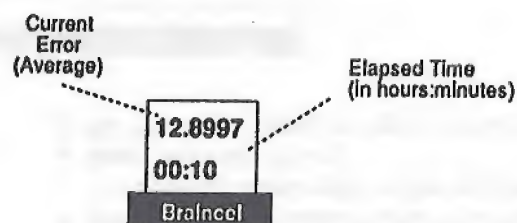


Figure 1.7 The two Train Expert boxes
We have filled in these boxes with the correct values for the Tutorial.



The training process is begun and Braincel is teaching your Expert. You can monitor learning progress by watching the Braincel icon at the bottom of your screen. The icon is color-coded to tell you its status. The text is green or red while training and black when the Expert is finished training or merely open. (Note: While training can be set to hours, minutes, and seconds, the icon displays only hours and minutes.)



As Braincel trains, watch the error in the Braincel icon. The error will change as the Braincel Expert learns. The overall trend of the error is downward as the Expert gets better at calculating the output. Often the error will increase for a time before decreasing. This may happen several times in a training session. This is normal.

Continue training until the error is approximately 5%
Add additional training time if needed to achieve a 5% error.

Testing the Expert's knowledge

Test the Expert to make sure it has learned from the cases. Testing is done with two sets of data: the Test Range and Predict Test Range.

- ✚ To test Expert training on the Test Range:
1. Select Expert Ask Expert. (If necessary, select Braincel from the Excel menu.)
 2. Select BCDATA.XLS as the worksheet.
 3. Add TEST_RANGE to the Selected Ranges box. Press OK.
 4. Compare the Output column to the Calculated Output column.

Loan Number	Monthly Income	Monthly Expense	Home Ownership	Year	Previous Job	Previous Address	Calculated Output
1	3000	2000	0	11	1	1	1
2	4000	1500	1	10	2	2	2
3	3000	1500	0	7	1	1	1
4	450	425	1	3	3	3	3
5	1000	3000	0	0.1	0.1	0.1	0.1
6	3000	2250	1	0	1	1	1
7	4000	1000	0	1	1	1	1
8	3500	2500	0	0.5	0.5	0.5	0.5
9	2500	1200	0	0	0	0	0
10	2500	2200	0	0	0	0	0
11	1200	1000	0	0.5	0.5	0.5	0.5
12	600	000	0	0.1	1	1	1
13	2500	300	10	3	10	3	3
14	3000	1000	20	0	10	10	10
15	2500	700	1	10	0	10	10

Figure 1.8 Testing the Expert's ability on the Test Range
By comparing the calculated output with the historical output, you can see that the Expert is doing very well on cases it has already seen and been trained on.

Note: If your own Tutorial.Net is not testing as well as the illustration above, continue training by selecting **Train Expert** and specifying a longer training time.

When the Expert is calculating well on the Test Range, you are ready to test its predicting ability on the Predict Test Range.

Testing the Expert's predicting ability

The Predict Test Range will test the Expert's predicting capability. You'll be showing the network data it hasn't been trained on, so there's no chance of it having memorized the output. Since the Predict Test Range is your own historical data, you'll have a historical output to compare the Expert's output against.

✚ To test the Braincel Expert's predicting ability:

1. Select Expert **Ask Expert**.
2. Select the worksheet to reset the Selected Ranges box.
3. Add **PREDICT_TEST** to the Selected Ranges box.
4. Compare the Output column with the Calculated Output column.

BCDATA.XLS											
Loan Repayment Ability											
Loan Number	Monthly Income	Monthly Expenses	Home Owned?	Years Present	Years Previous Job	Years Previous Address	Years Previous Address	Years Previous Address	Years Previous Address	Years Previous Address	Years Previous Address
1	2000	1500	1	10	5	3	2	1	1	1	1
2	2000	1500	1	10	5	3	2	1	1	1	1
3	2000	1500	1	10	5	3	2	1	1	1	1
4	2000	1500	1	10	5	3	2	1	1	1	1
5	2000	1500	1	10	5	3	2	1	1	1	1
6	2000	1500	1	10	5	3	2	1	1	1	1
7	2000	1500	1	10	5	3	2	1	1	1	1
8	2000	1500	1	10	5	3	2	1	1	1	1
9	2000	1500	1	10	5	3	2	1	1	1	1
10	2000	1500	1	10	5	3	2	1	1	1	1
11	2000	1500	1	10	5	3	2	1	1	1	1
12	2000	1500	1	10	5	3	2	1	1	1	1
13	2000	1500	1	10	5	3	2	1	1	1	1
14	2000	1500	1	10	5	3	2	1	1	1	1
15	2000	1500	1	10	5	3	2	1	1	1	1
16	2000	1500	1	10	5	3	2	1	1	1	1
17	2000	1500	1	10	5	3	2	1	1	1	1
18	2000	1500	1	10	5	3	2	1	1	1	1
19	2000	1500	1	10	5	3	2	1	1	1	1
20	2000	1500	1	10	5	3	2	1	1	1	1
21	2000	1500	1	10	5	3	2	1	1	1	1
22	2000	1500	1	10	5	3	2	1	1	1	1
23	2000	1500	1	10	5	3	2	1	1	1	1
24	2000	1500	1	10	5	3	2	1	1	1	1
25	2000	1500	1	10	5	3	2	1	1	1	1
26	2000	1500	1	10	5	3	2	1	1	1	1
27	2000	1500	1	10	5	3	2	1	1	1	1
28	2000	1500	1	10	5	3	2	1	1	1	1
29	2000	1500	1	10	5	3	2	1	1	1	1
30	2000	1500	1	10	5	3	2	1	1	1	1
31	2000	1500	1	10	5	3	2	1	1	1	1
32	2000	1500	1	10	5	3	2	1	1	1	1
33	2000	1500	1	10	5	3	2	1	1	1	1
34	2000	1500	1	10	5	3	2	1	1	1	1
35	2000	1500	1	10	5	3	2	1	1	1	1
36	2000	1500	1	10	5	3	2	1	1	1	1
37	2000	1500	1	10	5	3	2	1	1	1	1
38	2000	1500	1	10	5	3	2	1	1	1	1
39	2000	1500	1	10	5	3	2	1	1	1	1
40	2000	1500	1	10	5	3	2	1	1	1	1
41	2000	1500	1	10	5	3	2	1	1	1	1
42	2000	1500	1	10	5	3	2	1	1	1	1
43	2000	1500	1	10	5	3	2	1	1	1	1
44	2000	1500	1	10	5	3	2	1	1	1	1
45	2000	1500	1	10	5	3	2	1	1	1	1
46	2000	1500	1	10	5	3	2	1	1	1	1
47	2000	1500	1	10	5	3	2	1	1	1	1
48	2000	1500	1	10	5	3	2	1	1	1	1
49	2000	1500	1	10	5	3	2	1	1	1	1
50	2000	1500	1	10	5	3	2	1	1	1	1

Figure 1.9 Comparing outputs on the Predict Test Range
The accuracy on this range lets you know how well the Expert will perform on new cases.

Your own tutorial Expert should predict as well as the illustration in Figure 1.9. The Expert's calculated output should match the historical output. If it doesn't, check your Expert's error by selecting **Options Network Status**. If the error is higher than 5%, continue training by selecting **Train Expert** and specifying a longer training time.

Using the fully-trained Expert

Once the Expert has been fully-trained, you're ready to use its knowledge on new data.

Create an area to hold new records for Braincel to analyze

We've copied the column headings and placed directly below the training records. (Move down to ~R25C1.) Define a range to hold new loan application data and ask the Braincel Expert for its forecast on the loan repayment probability of the applicant.

✚ To define a New Record Range on BCDATA.XLS

1. Activate BCDATA.XLS, if necessary.
2. Go to cell R25C1.
3. Select R29C2:R29C10.

This range includes all input columns plus an empty column for Braincel to write its output into.

4. Using the Excel **Define Name** command, define this range as **NEW_RECORD**. (You may need to select **File Return To Excel** to use the Excel menu bar)

BCDATA.XLS											
Loan Repayment Ability											
Loan Number	Monthly Income	Monthly Expenses	Home Owned?	Years Present	Years Previous Job	Years Previous Address	Years Previous Address	Years Previous Address	Years Previous Address	Years Previous Address	Years Previous Address
1	2000	1500	1	10	5	3	2	1	1	1	1
2	2000	1500	1	10	5	3	2	1	1	1	1
3	2000	1500	1	10	5	3	2	1	1	1	1
4	2000	1500	1	10	5	3	2	1	1	1	1
5	2000	1500	1	10	5	3	2	1	1	1	1
6	2000	1500	1	10	5	3	2	1	1	1	1
7	2000	1500	1	10	5	3	2	1	1	1	1
8	2000	1500	1	10	5	3	2	1	1	1	1
9	2000	1500	1	10	5	3	2	1	1	1	1
10	2000	1500	1	10	5	3	2	1	1	1	1
11	2000	1500	1	10	5	3	2	1	1	1	1
12	2000	1500	1	10	5	3	2	1	1	1	1
13	2000	1500	1	10	5	3	2	1	1	1	1
14	2000	1500	1	10	5	3	2	1	1	1	1
15	2000	1500	1	10	5	3	2	1	1	1	1
16	2000	1500	1	10	5	3	2	1	1	1	1
17	2000	1500	1	10	5	3	2	1	1	1	1
18	2000	1500	1	10	5	3	2	1	1	1	1
19	2000	1500	1	10	5	3	2	1	1	1	1
20	2000	1500	1	10	5	3	2	1	1	1	1
21	2000	1500	1	10	5	3	2	1	1	1	1
22	2000	1500	1	10	5	3	2	1	1	1	1
23	2000	1500	1	10	5	3	2	1	1	1	1
24	2000	1500	1	10	5	3	2	1	1	1	1
25	2000	1500	1	10	5	3	2	1	1	1	1
26	2000	1500	1	10	5	3	2	1	1	1	1
27	2000	1500	1	10	5	3	2	1	1	1	1
28	2000	1500	1	10	5	3	2	1	1	1	1
29	2000	1500	1	10	5	3	2	1	1	1	1
30	2000	1500	1	10	5	3	2	1	1	1	1
31	2000	1500	1	10	5	3	2	1	1	1	1
32	2000	1500	1	10	5	3	2	1	1	1	1
33	2000	1500	1	10	5	3	2	1	1	1	1
34	2000	1500	1	10	5	3	2	1	1	1	1
35	2000	1500	1	10	5	3	2	1	1	1	1
36	2000	1500	1	10	5	3	2	1	1	1	1
37	2000	1500	1	10	5	3	2	1	1	1	1
38	2000	1500	1	10	5	3	2	1	1	1	1
39	2000	1500	1	10	5	3	2	1	1	1	1
40	2000	1500	1	10	5	3	2	1	1	1	1
41	2000	1500	1	10	5	3	2	1	1	1	1
42	2000	1500	1	10	5	3	2	1	1	1	1
43	2000	1500	1	10	5	3	2	1	1	1	1
44	2000	1500	1	10	5	3	2	1	1	1	1
45	2000	1500	1	10	5	3	2	1	1	1	1
46	2000	1500	1	10	5	3	2	1	1	1	1
47	2000	1500	1	10	5	3	2	1	1	1	1
48	2000	1500	1	10	5	3	2	1	1	1	1
49	2000	1500	1	10	5	3	2	1	1	1	1
50	2000	1500	1	10	5	3	2	1	1	1	1

Figure 1.10 The New Record Range
The highlighted cells comprise the New Record Range.

Show the Expert new data and get its analysis

In Figure 1.11 we're giving you a new loan application to present to your Braincel Expert. Enter this information into the New Record Range and then ask your Braincel Expert for its analysis.

After you've entered the information, your range should now look like Figure 1.12

Loan Application	
1. Monthly Income	2500
2. Monthly Expenses	1500
3. Home Owner? (Yes=1, No=0)	0
4. Years with Present Employer	3
5. Years with Previous Employer	2
6. Years at Present Address	3
7. Years at Previous Address	4
8. Number of Dependents	1

Figure 1.11 A new loan application
Put the answers on this application into the appropriate columns in the New Record Range.

	2	3	4	5	6	7	8	9	10	11	12
26				Yr at	Yr at	Yr at	Yr at				
27	Monthly	Monthly	Home	Present	Previous	Present	Previous	No. of	Calculated		
28	Income	Expenses	Owner?	Job	Job	Address	Address	Depend	Output		
29	2500	1500	0	3	2	3	4	1			
30											
31											
32											

Figure 1.12 The New Record Range filled in
Place the responses from the loan application in Figure 1.11 into the cells in the New Record Range, as shown above.

- ✚ To ask the Expert on the New Record Range:
1. Display the Braincel menu.
Select **Braincel** from the Excel Menu, if necessary.
 2. Select **Expert Ask Expert**.
 3. Select **BCDATA.XLS** to clear the Selected Ranges box.
 4. Add **NEW_RECORD** to the Selected Ranges box.
Press **OK**.
 5. When the Expert is finished processing, go to **R29C10**, the cell in the New Record Range where the Expert's output will appear.

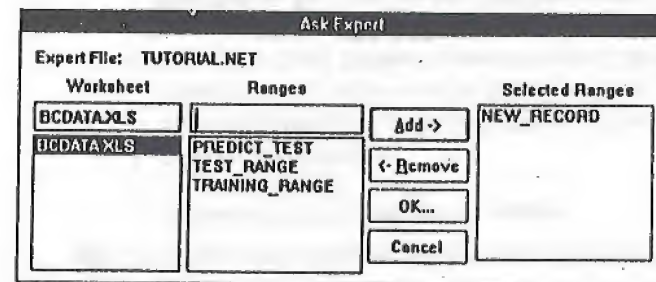


Figure 1.13 Asking the Expert on the New Record Range

	2	3	4	5	6	7	8	9	10	11	12
26				Yr at	Yr at	Yr at	Yr at				
27	Monthly	Monthly	Home	Present	Previous	Present	Previous	No. of	Calculated		
28	Income	Expenses	Owner?	Job	Job	Address	Address	Depend	Output		
29	2500	1500	0	3	2	3	4	1			
30											
31											
32											

Figure 1.14 The Expert's forecast

Just like the human loan officer, the Braincel Expert assigns loan risk on a scale of 1 to 5, 1 meaning very poor loan repayment probability and 5 meaning excellent loan repayment probability.

The Expert is linked to the New Record Range. You can enable a hot-link between the Expert and this range by selecting **Enable Ask Link**. Now, if you change any value in any of the cells in this range, the Expert will reevaluate the application. For example, select **Enable Ask Link** then change the value in the **Monthly Income** cell from 2500 to 3500 and see how that affects the Braincel Expert's prediction. The Expert will update its prediction automatically. You can disable this hot-link by selecting **Disable Ask Link**.

Now you're ready to move on to creating your own applications.

Creating a basic Braincel Expert

This section is designed to provide details on the basic process: Excel worksheet setup, creating, training and using the Expert. It is a supplement to the Tutorial.

This section shows you how to set up and use Braincel with the least amount of work using the **Auto Expert User Mode** (the default User Mode determined in the **Options Setup** box). Braincel will configure and monitor training automatically.

After using **Auto Expert mode**, experienced neural net users should refer to the **Variations** section to learn about the **Professional User Mode** and its more advanced features.

Quick Checklist for creating a Braincel Expert

- Choose a problem to solve or something to forecast.
- Collect historical data as examples for the network to learn from (or use a human expert to create cases).
- Load the data into an Excel worksheet.
- Divide the data into inputs and outputs.
 - Each input or output into a separate column, one example per row.
 - Minimum and maximum values for each column as the last two rows. (These rows can be blank; Braincel will fill them.)
- Define three data sets as named ranges (with the Excel Define Name command) See Diagrams on the next two pages.
 - Training range—90% of your records with min/max rows.
 - Test Range—the training range without the historical output columns and including empty columns for the expert to write its calculations in to (one empty column for each output).
 - Predict Test Range—the 10% of the total records withheld from training without the historical output columns and with empty columns for the expert to write its calculations in to.
- Open a New Expert.
- Train the Expert with the Training Range.
- Test the Expert on the Test Range (with the Ask Expert command).
- Test the Expert on the Predict Test Range (with the Ask Expert command).
- Use the Expert on new data by defining a range to hold the new data then using the Ask Expert command.

Diagrams for the Quick Checklist

The following diagrams are not hard and fast rules; they are schematic representations. Details on worksheet organization options are in the pages following.

	1	2	3	4	5	6	7	8	9
1								Braincel	Braincel
2		Input 1	Input 2	Input 3	Input 4	Output 1	Output 2	Output 1	Output 2
3	Record 1	xxx	xxx	xxx	xxx	xxx	xxx		
4	Record 2	xxx	xxx	xxx	xxx	xxx	xxx		
5	Record 3	xxx	xxx	xxx	xxx	xxx	xxx		
6	Record 4	xxx	xxx	xxx	xxx	xxx	xxx		
7	Record 5	xxx	xxx	xxx	xxx	xxx	xxx		
8	Record 6	xxx	xxx	xxx	xxx	xxx	xxx		
9	Record 7	xxx	xxx	xxx	xxx	xxx	xxx		
10	Record 8	xxx	xxx	xxx	xxx	xxx	xxx		
11	Record 9	xxx	xxx	xxx	xxx	xxx	xxx		
12	Record 10	xxx	xxx	xxx	xxx	xxx	xxx		
13	Min	xxx	xxx	xxx	xxx	xxx	xxx		
14	Max	xxx	xxx	xxx	xxx	xxx	xxx		

Basic Worksheet Setup

Each input or output is in a continuous column. Minimum and maximum values are the last two rows. The outputs are the last columns.

	1	2	3	4	5	6	7	8	9
1								Braincel	Braincel
2		Input 1	Input 2	Input 3	Input 4	Output 1	Output 2	Output 1	Output 2
3	Record 1	xxx	xxx	xxx	xxx	xxx	xxx		
4	Record 2	xxx	xxx	xxx	xxx	xxx	xxx		
5	Record 3	xxx	xxx	xxx	xxx	xxx	xxx		
6	Record 4	xxx	xxx	xxx	xxx	xxx	xxx		
7	Record 5	xxx	xxx	xxx	xxx	xxx	xxx		
8	Record 6	xxx	xxx	xxx	xxx	xxx	xxx		
9	Record 7	xxx	xxx	xxx	xxx	xxx	xxx		
10	Record 8	xxx	xxx	xxx	xxx	xxx	xxx		
11	Record 9	xxx	xxx	xxx	xxx	xxx	xxx		
12	Record 10	xxx	xxx	xxx	xxx	xxx	xxx		
13	Min	xxx	xxx	xxx	xxx	xxx	xxx		
14	Max	xxx	xxx	xxx	xxx	xxx	xxx		

The basic Training Range

Should be approximately 90% of your records. Includes all inputs, the min/max value rows and the actual, historical outputs.

Diagrams for the Quick Checklist cont....

	1	2	3	4	5	6	7	8	9
1									
2		Input 1	Input 2	Input 3	Input 4	Output 1	Output 2	Braincel	Braincel
3	Record 1	1000	1000	1000	1000	1000	1000		
4	Record 2	1000	1000	1000	1000	1000	1000		
5	Record 3	1000	1000	1000	1000	1000	1000		
6	Record 4	1000	1000	1000	1000	1000	1000		
7	Record 5	1000	1000	1000	1000	1000	1000		
8	Record 6	1000	1000	1000	1000	1000	1000		
9	Record 7	1000	1000	1000	1000	1000	1000		
10	Record 8	1000	1000	1000	1000	1000	1000		
11	Record 9	1000	1000	1000	1000	1000	1000		
12	Record 10	1000	1000	1000	1000	1000	1000		
13	Min	1000	1000	1000	1000	1000	1000		
14	Max	1000	1000	1000	1000	1000	1000		

The Test Range

Includes all inputs in the Training Range records plus an empty column for each historical output. Braincel will write its calculations in the empty columns. The min/max rows are not included.

	1	2	3	4	5	6	7	8	9
1									
2		Input 1	Input 2	Input 3	Input 4	Output 1	Output 2	Braincel	Braincel
3	Record 1	1000	1000	1000	1000	1000	1000		
4	Record 2	1000	1000	1000	1000	1000	1000		
5	Record 3	1000	1000	1000	1000	1000	1000		
6	Record 4	1000	1000	1000	1000	1000	1000		
7	Record 5	1000	1000	1000	1000	1000	1000		
8	Record 6	1000	1000	1000	1000	1000	1000		
9	Record 7	1000	1000	1000	1000	1000	1000		
10	Record 8	1000	1000	1000	1000	1000	1000		
11	Record 9	1000	1000	1000	1000	1000	1000		
12	Record 10	1000	1000	1000	1000	1000	1000		
13	Min	1000	1000	1000	1000	1000	1000		

The Predict Test Range

Contains the 10% of your records that the Expert wasn't shown during training. (These records should be randomly selected.) Include all input columns plus an empty column for each output.

What can I do with Braincel?

Braincel is great for forecasting and building expertise. It works best with problems that share the following characteristics:

- The cost of developing rules is prohibitive.
- The formulas are constantly changing.
- Lots of data is available.

Within this framework there are two approaches to a problem with Braincel: 1) Imitating human expertise where the rules for solution are unknown, like the loan expert in our tutorial; 2) Information mining—looking for relationships within data. This approach is best used by people with expertise in the field they are studying but want to look at the data their data in new ways.

Examples:

- "How many widgets will I need in my inventory?"
- "Who is the mostly likely winner of today's horse race?"

Braincel should not be used when the formulas for a decision are already known and are fairly static. For instance, you shouldn't use a neural network to calculate an 8% sales tax. It's much easier to multiply by 0.08 than to teach the network to multiply with accuracy. Neural network experts should be used to solve problems for which it is difficult to formulate a procedural software solution or if the procedure is likely to change frequently.

Another factor to consider is how you want to present the problem to the neural network. If you have a problem with several outputs, you could break the problem into several networks, each with one output. Additionally, you can use the output of one or more networks as the inputs to another network. For example, for the tutorial loan Expert, we could have constructed one network that estimates the stability of a loan applicant, based on employment record and housing record. Another network estimates the amount of money an applicant can pay, based on assets and expenses. The outputs of these two

networks are inputs to a third network that evaluates the total loan.

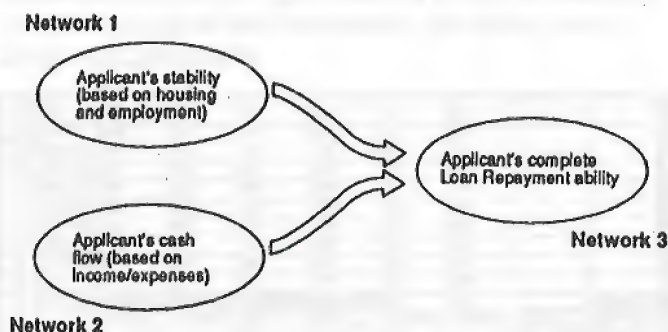


Figure 2.1 Using the outputs of Experts as inputs to another Expert

This example shows an alternative configuration for the tutorial loan Expert. Two networks assess specific risks. Their outputs are used in a third Expert that assesses the total risk.

Braincel is good for a large range of problems from forecasting the stock market, forecasting sales, filtering mailing lists, to predicting horse races.

Example: Stock and Commodities

Use the factors that an expert in the field of forecasting futures uses to analyze. You can also include a couple of factors whose influence you're unsure of. If these factors are immaterial, the network will minimize their effect in calculations.

Example: Mailing list filter

Keep statistics on the efficacy of past mailings, as many factors as you have access to (Hint: you could store this information in a dBASE file and link it to the worksheet via Excel's Q+E™.) Then train the network to predict, for example, how accurate a mailing will be, based on the success or failure of test mailings.

The neural network is good at this because you can include factors that may or may not be immediately obvious, such as

economic factors. Braincel's calculating power allows you to add these factors without worry: if Braincel finds that they're important, it will use them. If they aren't important, Braincel will minimize their influence in its calculations.

Example: Scientific or Technology Applications

Braincel is an excellent tool because you can use it to minimize expensive and time-consuming experimentation by pinpointing likely success.

For example, let's say we're developing a new type of plastic for a customer. Our customer will request various physical properties for the plastic.

We create a neural network Expert. We use as inputs the physical properties of plastics made for previous customers. The variables used in making those plastics are the outputs. We train the Expert on those past cases. Then we ask it which variables to use to get the new physical properties. The plastic will be made at a much lower cost since fewer actual experiments must be done. The neural network has pointed the way to making the new plastic.

Other applications include analyzing physical properties of metals and drugs. It's an excellent tool for making discoveries in engineering and scientific disciplines.

What kinds of data should I include as inputs?

In general, you should include any information that would be helpful to a human decision-maker. The human decision-maker is known as a domain expert. This person has knowledge in the field under consideration. For example, when constructing a loan approval Expert like the one created in the tutorial, you should be in contact with a human loan approval officer to direct you towards the type of information important to such a decision. This person would be the domain expert.

Creating a Basic Braincel Expert

For example, in the tutorial we created a loan Expert. The loan Expert was given eight inputs even though a loan approval decision could probably be made on the basis of just two questions: Monthly Income and Monthly Expenses. If a person's expenses are greater than his income, he shouldn't be given a loan. But the human loan expert who provided the historical data looked at several other factors. These factors helped the expert assess each applicant's loan repayment probability more completely. For example:

- How long has the applicant lived at her current address?
- How long has she worked at the same job?

These are important to the human expert, although she may not be able to explain why. Therefore, we gave all the information to the Braincel Expert.

When you are developing your own Expert, include as many factors as you consider necessary. Humans often overlook their own decision-making criteria, so it's better to give the program more inputs rather than fewer. Too many unimportant inputs, however, may increase training time unnecessarily.

Additionally, you should exclude factors that measure the same feature. For example, if you have "Gross National Product (GNP)" as one input in a financial model, you shouldn't have "GNP in 1982 dollars" as another input. Though they're different factors, they're really measuring the same thing. However, one may be more helpful to the neural network than the other. You could try making two networks, one with GNP and one with GNP in 1982 dollars. One may be more accurate than the other. This type of data manipulation is called preprocessing. There are many types of preprocessing. Some options are discussed in the Variations—Improving your Expert's accuracy.

Braincel will accept data in many forms: as numbers, formulas or, as we explain in the Variations section, text. You can have a combined total of 256 inputs and outputs.

Details on organizing your data in Excel

There are three basics to consider when organizing your data:

1. Each input or output must be in a separate column.
2. Each column must have space for a minimum and maximum value as the last two rows.
3. Define three data sets as named ranges: the Training Range, the Test Range and the Predict Test Range.

Inputs and outputs in separate columns

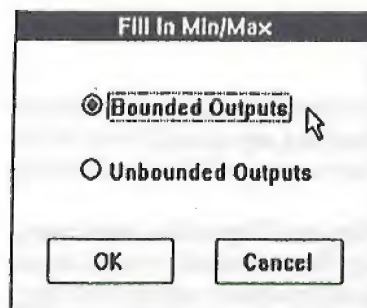
Each column of data must be continuous in the defined range. For example, an input can't consist of rows 1–50 and 65–70. The input should be rows 1–70. It's ok for rows 51–64 to be blank as long as they are included in the defined range.

The outputs must be placed as the right-most columns when using a single range to define the Training Range. (You may use more than one range to define the total Training Range. In this case, the outputs need not be the right-most column or even next to each other. This is explained in Variations—Using Multiple Ranges, pp 56-59.)

Minimum and maximum values

Each column of data must have a minimum and maximum value at the bottom in the last two rows. Braincel is using these min and max values to determine what it can expect to see in the column.

You can set your own minimum and maximum values or Braincel will calculate min/max values for you. The inputs are given a minimum value slightly less than the actual minimum and a maximum value slightly greater than the actual maximum. The range for an output depends on what type of output it is. If the problem is a classification problem (output is basically Yes/No or some variant) or the output should never fall outside the cases presented in training, then we call this a bounded output. If the output is a number that could exceed the scale presented in training, then we call this an unbounded output. An example of an unbounded output is a raw stock price. A bounded output would be a buy/sell signal on the stock.



Min/max autofill
Braincel will fill your min/max rows depending on which type of output you have, bounded or unbounded.

If you choose to set your own min/max values, use common sense in your choices. For example, in the tutorial Loan Experts, there was a "Number of Dependents" input. We could give this column a minimum value of 0 and a maximum value of 10. It's impossible to have fewer than zero dependents and very few people in the real world have more than 10 dependents. (No applicant in the training range had more than 5). Setting a maximum value of 100 would cause scaling problems for the Braincel Expert because it would consider having 9 dependents almost the same as having 1. This wouldn't show the network how different having 9 dependents is from having 1 dependent.

Three necessary data sets: The Training Range, Test Range and Predict Test Range

Braincel needs to see your data in three different sets as part of its learning process.

The largest set is the **Training Range**. It should include approximately 90% of your data records. (You leave out 10% of the records for testing later in the training process.) All input and output columns, with their minimum and maximum rows are included. The Training Range needs to be large enough to show the Expert-in-training as many possible combinations of inputs as your machine memory and time permit. The network is learning by example, so the more examples you have, the better.

The second set is the **Test Range**. This range is used to see how accurate the Expert is on specific records in the training range. The Test Range forces the Expert to write its calculations for each record onto the spreadsheet so you can compare the calculations to the actual historical output. Now you can see how the network is performing on individual cases. This gives you a more accurate representation of the Expert's performance than the average error percentage you see in the Braincel icon. Since the error percentage in the icon is an average, it deemphasizes the records that the Expert is not learning well.

The Test Range includes all input columns plus an empty column for each historical output. The minimum and maximum values are not included. You can test on all training records or on a smaller subset.

The third set is the **Predict Test Range**. This is the 10% of the records that you withheld from training. Include the input columns for these records, plus an empty column for each output. The Braincel Expert will write its calculations into these empty columns.

The Expert-in-training hasn't seen these records. Testing on this range will give you a good idea of how accurate the Expert will be when it's given new data. Since this is historical data, you know what the correct output should be. Compare the historic output with the Expert's calculated output. This test will give you a good idea of how well the Expert will predict on new data.

When choosing records for the Predict Test Range, select records that represent the full spectrum of your data. For instance, in the tutorial loan Expert, we selected two records representing two different loan officer responses out of five possible. The tutorial was based on a tiny database. In your own application, use the Excel database functions to select a representative sample. This will show you whether or not the Expert is accurate throughout the database.

These three data sets do not have to be single ranges. They can each be made up of several named ranges. See *Variations—Using Multiple Ranges* for information.



Details on the Training/Testing process

Braincel Expert learns the relationships in the data during the training and testing processes. Braincel self-monitors most of the learning process. The dynamic internals of the neural network (for example, learning rate) are controlled and the physical internal configuration is set by Braincel itself. You can override this self-monitoring. See Variations—Working in Professional User Mode.

Just like teaching a human, Braincel requires a bit of supervision while learning. Primarily, the user determines how accurate the Expert should be on the training data. This accuracy is determined by the error specified during the Train Expert command. Also, the user specifies how long the Expert should train.

Determining an acceptable error

Different problems require different levels of accuracy. You want the expert to be accurate, but not so accurate that it has memorized the training data, causing it to perform poorly on new cases. The error in the icon is an average of the errors for each record. The error for an individual record is calculated as follows:

$$\text{Error} = \frac{\text{Average}(\text{Hist. Output} - \text{Expert Calculated Output})}{\text{Standard Deviation of Hist. Output}}$$

For example, the tutorial loan Expert had a min-max range of 1–5. If the network calculated a 4 and the historical output was a 3, the error for that record would be 25%.

It's best to interrupt training occasionally and test the Expert against both the training data and the unseen testing data; that is, the Test Range and the Predict Test Range. In this way, you'll see how accurate the Expert will be when asked to analyze new data. As long as the Expert continues to improve performance on the Predict Test Range, continue training to a lower error. As soon as performance worsens, stop training.

You can automate this training and testing process in two ways:

- 1) Use Automated Best Net Search (details pp 41-47); 2) train on Unseen Data in Professional User Mode (details p. 70).

Determining an appropriate training time

It is very difficult to determine in advance what an appropriate training time will be. In general, the more records you have, the longer the network will need to analyze the records. Also, the more inputs and outputs in each individual record, the more time the network will need.

You may specify a training time of up to 99 hours 59 minutes and 59 seconds. If you need more time than this, you can set the Expert for additional cycles after the previous cycle is complete.

Making sure the Braincel Expert is learning

It is necessary to test the Expert during training. Only by testing will you determine how accurate the Expert truly is. The error percentage in the icon refers to the average difference scaled for the standard deviation of the output.

This doesn't necessarily give you a true picture of how accurate the Expert is. For instance, we recommended that you test your loan Expert after it has achieved a 10% percent error. A 10% error may sound very high to you, but in fact it may be desirable, depending on your application. The error doesn't indicate how many times the network gives a correct answer. Consider the following example.

Example:

You have constructed a rainfall-predicting Expert in which an output of 1 means rain; 0 means no rain. The Expert calculates 0.75 for a particular day. 0.75 is closer to 1 than it is to 0, so the Expert predicts rain. If the weather is rainy that day, the Expert has given the correct answer. BUT its error is listed at 25%.

It's not necessarily desirable to train to an extremely low error, like 1%; if the error is too low, the network will have trained itself too well. It will have memorized every record individually and not be able to generalize its knowledge to records it has

never seen before. You must train and test, adjusting the error to get the results on the Predict Test Range that are adequate for your needs.

Generally, the Expert will not perform as well on the Predict Test Range as it did on the Test Range; it has never seen the data in the Predict Test Range before. You can expect the error percentage on the Predict Test Range with a well-trained Expert to be up to 10 points higher than it was on the Test Range.

When you're satisfied with the Expert's accuracy on the Predict Test Range, training is completed. Your Expert is ready to examine new data.

Note: When creating your own Braincel Experts, if you are unsatisfied with the network's accuracy on the Predict Test Range but were satisfied with the network's performance on the Test Range, refer to Troubleshooting—Poor Performance on Predict Test Range.

Can I use other Windows applications while training?

You can enable task switching in Options Setup. This will allow you to use ALT-TAB to interrupt Braincel and switch to the Program Manager and hence other applications. However, Braincel will halt training until Excel is again the current application.

Notes on using the fully-trained Expert

Your Expert is ready to be used on new data. You can analyze this data by defining a New Record Range and putting the new data into it. The inputs must be entered in the same order they were presented for training. If you don't follow the training order, the calculated result will be unpredictable.

What does the New Record Range include?

The New Record Range includes columns to hold the inputs and an empty column to hold each calculated output. There are no minimum and maximum value rows. We recommend that you paste a copy of the same headings that you used in your training and testing ranges above your New Record Range. This will make it easier for you to keep your inputs in the same order as

you presented them in training.

The link between the network file and the worksheet

After you've analyzed the new data with the Ask Expert command, the Expert file can be hotlinked to that range. You can enable a hot-link by selecting *Enable Ask Link* in the Expert menu. With this hot-link, the Braincel Expert will automatically recalculate the linked range any time a value in one of the linked cells changes.

How can I use the Expert on a different worksheet?

You can overwrite the worksheet and ranges in the *Ask Expert* box and start filling the Selected Ranges box with ranges from a different worksheet.

All worksheets open in Excel will appear in the Worksheet list box in *Ask Expert*.

Modifying Expert training

You can modify your Expert's knowledge by giving it more data and retraining the Expert. For instance, the human loan approval officer could keep recording his or her cases and giving that information to the Braincel loan Expert. So, if the human modified his or her decision-making criteria, the Braincel Expert would see this pattern and modify its decision making.

Note: When you update your Expert's knowledge, you must add the new cases to the original training range then start training again. Your training range will increase in size and the training time may also increase.

Using Evolver

13

WHARTON REPROGRAPHICS

Introduction

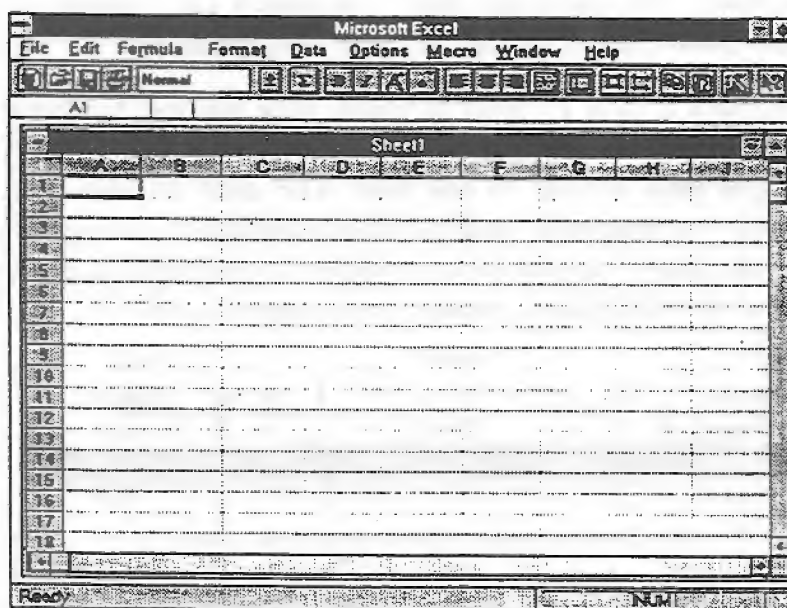
In this chapter, we will take you through the entire Evolver process step by step. We will start by opening a simple spreadsheet model and defining the problem to Evolver. Then Evolver will optimize the problem, searching for the best solution. We will also discuss many of Evolver's special features. For additional information about any topic, see Chapter 5: Reference.

Opening the Tutorial

If you do not have Evolver installed on your hard drive, please refer to the installation section of Chapter 1: Getting Started and install Evolver before you begin this tutorial.

1. Open Microsoft Excel for Windows.

Excel will automatically open a blank spreadsheet titled "Sheet1."



Excel presents you with an empty worksheet.

2. Close the blank sheet.

3. Under the "File" menu, select "Open."

Evolver User's Guide

4. Open the "tutorial.xls" file located in the "examples.ev2" subdirectory. This directory is inside the "xlstart" subdirectory, which is located in your Excel directory.

The "tutorial.xls" worksheet appears as below:

Worker	1	2	3	4	5	6	7	8	9	10	Chosen Task #
Steve	6	5	4	1	10	10	4	2	3	9	1 = 6
Robert	10	9	10	2	8	4	7	1	10	3	2 = 9
Susan	6	0	7	5	8	5	5	10	9	7	3 = 7
Greg	0	5	2	10	2	4	1	2	5	1	4 = 10
Austin	8	5	8	0	9	8	3	9	6	5	5 = 9
Joe	10	9	8	9	10	6	10	8	8	10	6 = 6
Frank	2	8	7	7	0	10	2	5	3	8	7 = 2
Jule	2	5	0	10	0	4	4	9	7	2	8 = 9
Kelly	6	9	3	9	7	6	5	3	2	0	9 = 2
Michael	3	2	3	10	2	0	7	0	4	6	10 = 6

Total Score: 66

The "tutorial.xls" spreadsheet describes a resource allocation problem where each worker must perform one task.

This worksheet models a common problem involving resource allocation. In this problem, you have 10 workers to perform 10 tasks. Each worker's ability to perform each task is rated on a scale of 0 to 10 (0= cannot do the task, 10= perfect at the task). The challenge here is to match each worker to a task so that the overall productivity of the workers is maximized.

The model provides a 10 x 10 grid in which each worker has been rated for each task. The "Chosen Task" column (column N) to the right of the grid arbitrarily assigns each worker to one task. The next column over (column P) checks what task was assigned, and enters each worker's rating for that task. Finally, the total "score" of the entire solution in cell P15 is the sum of adding up all of the individual ratings (see screen below).

Worker	1	2	3	4	5	6	7	8	9	10	Chosen Task #
Steve	6	5	4	1	10	10	4	2	3	9	1 = 6
Robert	10	9	10	2	8	4	7	1	10	3	2 = 9
Susan	6	0	7	5	8	5	5	10	9	7	3 = 7

In the model, Robert is assigned task 2, at which he rates a "9".

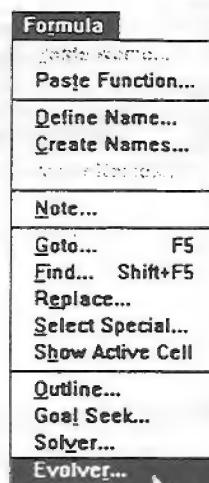
5. Click on cell N5 (currently a 2), and change the value to 6.

When you enter the new value, the model sees that Robert rates a "4" at task #6 and recalculates the Total Score to "61."

6. IMPORTANT: Change the value of cell N5 back to 2.

Although this problem may seem simple, there are over 3.6 million possible ways to assign these workers to their tasks. If we added just one more constraint to the problem (e.g. if certain tasks required 2 workers, or some tasks required other tasks to be completed first), the problem's complexity would increase exponentially.

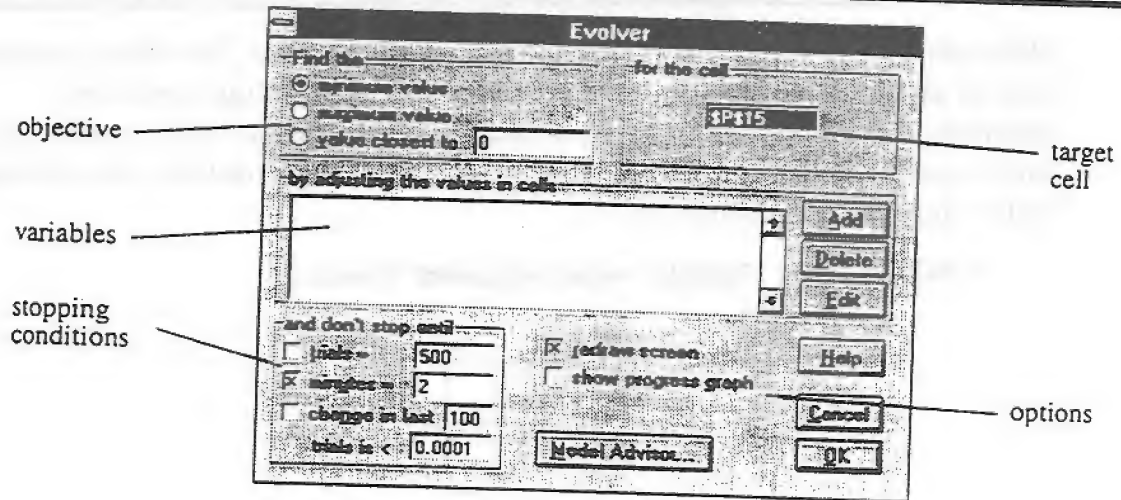
7. Pull down the "Formula" menu and select "Evolver."



NOTE: If Evolver is not available, check to make sure that you have installed it correctly. The file "evstub.xla" adds the "Evolver" item to the bottom of Excel's "Formula" menu. If this file is not in your "xlstart" directory, you must manually open this file from within Excel. For more installation information, see Appendix B at the back of this manual.

When selected, the "Evolver" command takes about 10 seconds loading in the "Evolver.xla" file and all of the Evolver solving methods. Evolver will only load once, and will remain available as long as Excel is open.

Selecting Evolver opens the following Evolver main dialog:



Describing The Problem

The Goal

The Evolver main dialog is designed so users can describe their problem in a simple, straightforward way. In our tutorial example, we are trying to find the combination of workers to tasks that produces the maximum overall "score."

8. Set the "Find the..." setting to "maximum value."



9. Check that cell \$P\$15 is entered as the target cell.



All information in Evolver can be entered in the dialogs in two ways: you may type the reference into the field with absolute coordinates (\$B\$4, not B4), or with your cursor in the selected field, you may click on the cell(s) directly with the mouse. To select the spreadsheet cells underneath the dialog, drag the dialog to one side.*

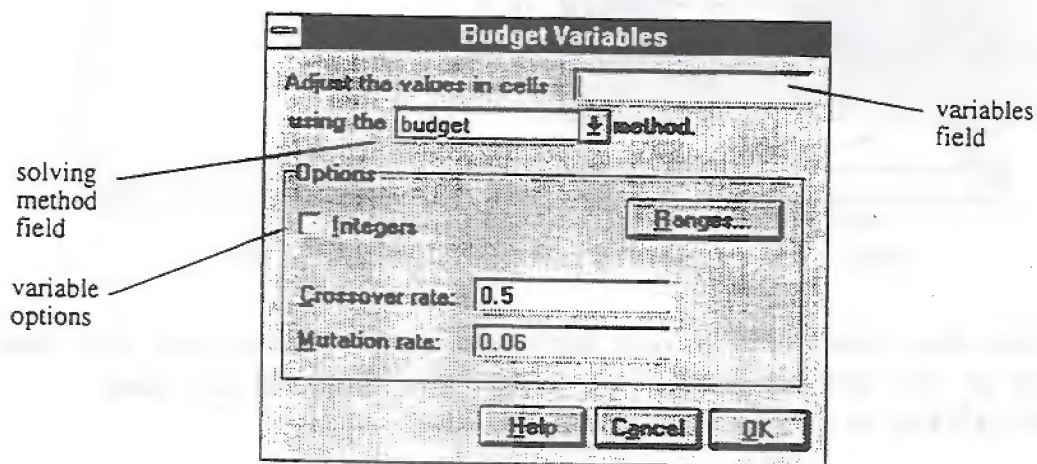
* To drag and move any window (or dialog), click on the title bar across the top of the window and hold down the mouse button while you drag it to a new location.

The Variables

To complete the description of the problem, you must specify the location of the variables that Evolver can adjust as it searches for a better solution. Evolver can handle an unlimited number of variables. All variables are added and edited one block at a time, through the variables dialog.

10. Click the "Add" button.

When you "Add" or "Edit" variables, Evolver will present the following variables dialog box:



The variables dialog prompts the user to select the variables and choose how they should be treated.

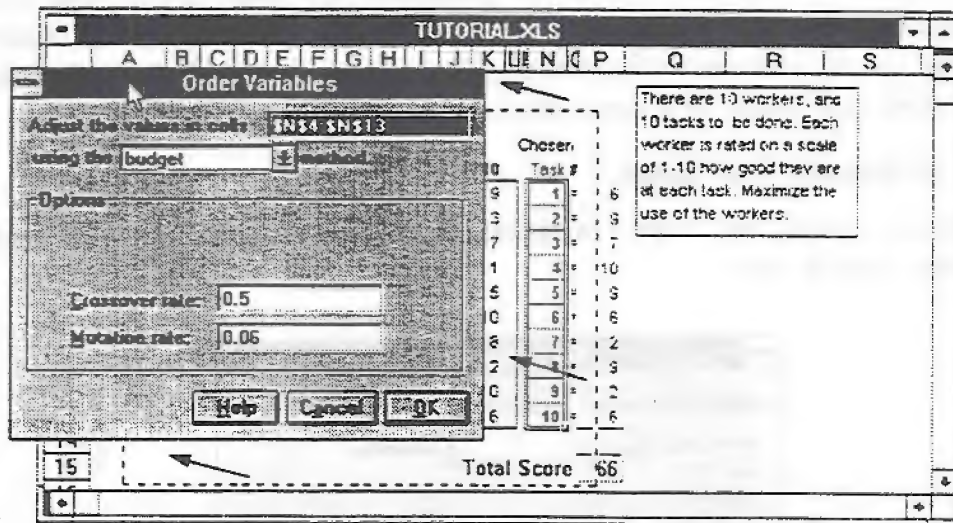
You will enter a block of variable cells in the variables field, and then specify a solving method to be used on those variables. Different types of variables are handled by different solving methods. The "recipe" solving method, for example, treats each variable as an ingredient in a recipe; each variable's value can be changed independently of the others'. In contrast, the "order" solving method swaps values between the adjustable cells, trying out different permutations of the original values*. In our tutorial example, we have a block of variables in column N that we want Evolver to adjust.

NOTE: When you are modeling your problem in Excel, remember to group together all of the variables you want to be adjusted in a certain way. This way, each group or block of variables can then be defined by the upper left cell and the lower right cell. A separate sub-problem should be added for each distinct block of variables.

11. Drag the variables dialog to the left.

* For more information, see the "Solving Methods" section in [Chapter 5: Reference](#).

12. With the cursor in the variables field, drag the cursor across column N4-N13 (see dialog below).



Drag the dialog to the side, then with the cursor placed in the variables field, drag across the block of variables to enter the cells.

As you drag across cells in your worksheet, your variables field will show the block of cells being entered. Your variables field should now read `N4:N13` as in the screen above.

The Solving Methods

In Evolver, different types of variables are handled by different *solving methods*. Now that we have chosen our variables for this problem, the second part of the variables dialog involves choosing which solving method should be used.

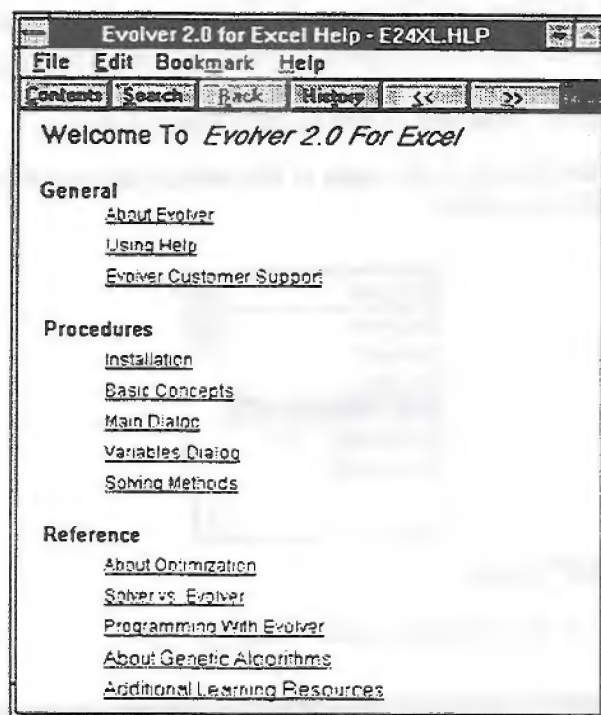
Let us say that you do not understand which solving method to use. If you have any questions regarding Evolver, you may want to use the Evolver on-line help file.

Using Help

Both the Evolver main dialog and the Evolver variables dialog offer Help buttons. You can also access Evolver Help at any time (while Evolver dialogs are visible) by pressing the F1 key on your keyboard.

13. Push the "Help" button, or press the F1 key on your keyboard.

This will call up the following Evolver Help window:



The Evolver Help window is just a mouse click away.

The Help window contains information indexed by topic. You can select any topic by clicking on it, or by clicking on the "Search" button to lookup a specific term. By navigating through the Help window, you can learn more about solving methods, and find examples of where to use which method.

NOTE: The Evolver Help system is designed like most standard Windows help systems. If you are new to using this type of Help system, refer to your Excel User's Guide, or select the "Using Help" topic.

14. To learn more about solving methods, click on the "Solving Methods" term.

15. Read about the "Recipe" and "Order" solving methods.

You learn that these two solving methods are the most popular, and that they can be used together to solve complex combinatorial problems. Specifically, you learn that the "recipe" solving method treats each variable as an ingredient in a recipe, trying to find the "best mix" by changing each variable's value independently. In contrast, the "order" solving method swaps values between variables, shuffling the original values to find the "best order."

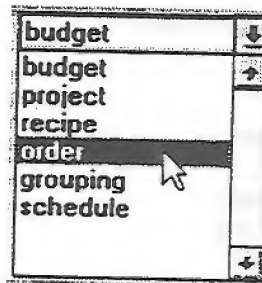
In this problem we are looking for the best way to shuffle the existing variables, so the "order" solving method should be used. If you are still unsure about which solving method to use on your variables, refer to the

Evolver User's Guide

Solving Method section of [Chapter 5: Reference](#) for a complete description of each method and the specific options that accompany them.

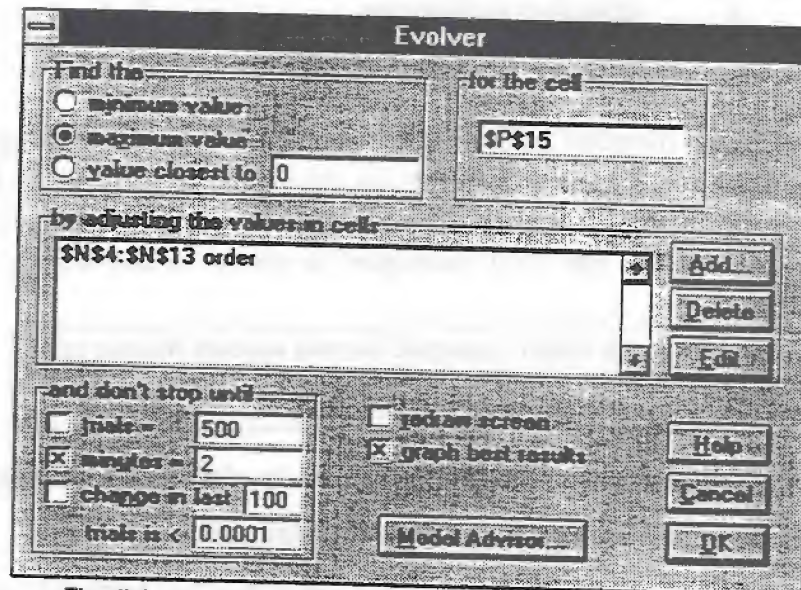
16. Close Evolver Help and return to the variables dialog.

17. Click on the button to the right of the solving method field, then select the "order" solving method.



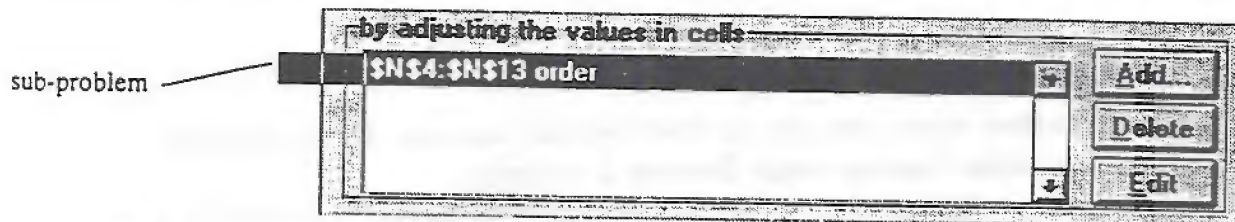
18 Push the "OK" button.

This returns you to the Evolver main dialog.



The dialog now describes what needs to be done, and how to do it.

You will notice that the variables you selected, along with the solving method to use on those variables, are now listed in the variables field of the Evolver main dialog. Each set of variables, and the settings for how they should be treated, is a *sub-problem* (see [Chapter 5: Reference](#)).



If there were additional variables in this problem, we would continue to add sub-problems for each set of variables. In Evolver, you may create an unlimited number of sub-problems.

19. Click the "Add" button again.

A new Evolver variables dialog will appear which allows you to choose new variable cells. In this problem, however, we only have one sub-problem.

20. Click "Cancel" to return to the Evolver main dialog.

Later, you may want to check the variables or change some of their settings. To do this, simply click on the sub-problem you would like to inspect (the sub-problem will appear highlighted in inverse), and click the "Edit" button. You may also select a selected sub-problem and delete it by pushing the "Delete" button.

The Stopping Conditions

Evolver can run as long as you wish. The stopping conditions allow Evolver to automatically stop when either: a) a certain number of scenarios or "trials" have been examined, b) a certain amount of time has elapsed, or c) no improvement has been found in the last n scenarios.

You can select any combination of these three stopping conditions, or none at all, through the main Evolver dialog. You can also stop Evolver manually by pressing the "Esc" key while Evolver is running.

<input checked="" type="checkbox"/> trials	<input checked="" type="checkbox"/> minutes	<input checked="" type="checkbox"/> change in last
This option sets the number of trials that you would like Evolver to run. In each trial, Evolver evaluates one possible solution or scenario.	Evolver will run for the specified amount of time until it stops.	This stopping condition is the most popular because it keeps track of the improvement and allows Evolver to run until it is no longer finding better solutions.

21. Turn on the minutes option only, and change the number of minutes to 3.



The Screen Options

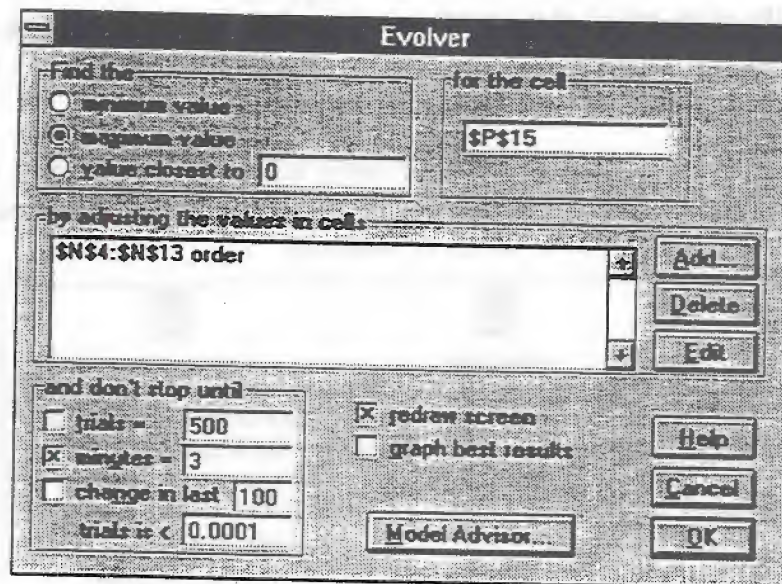
When Evolver runs, you can set two viewing options. These settings determine what you see while Evolver is running.

<input checked="" type="checkbox"/> redraw screen	<input checked="" type="checkbox"/> graph best results
This option redraws the screen after each calculation, allowing you to see Evolver adjusting the variables and calculating the output. We suggest this option be turned on while you are learning Evolver, but later turned off to increase Evolver's speed.	When this option is selected, Evolver will build a graph and update it after every 20 scenarios plotting the solution so that users can see how their problem is progressing as Evolver is optimizing.

22. Select the "redraw screen" option, and de-select the graph best results screen.

☒ redraw screen
☐ graph best results

The following screen illustrates what your Evolver main dialog should look like at this point:



This is the Evolver main dialog completely filled out.

Running Evolver

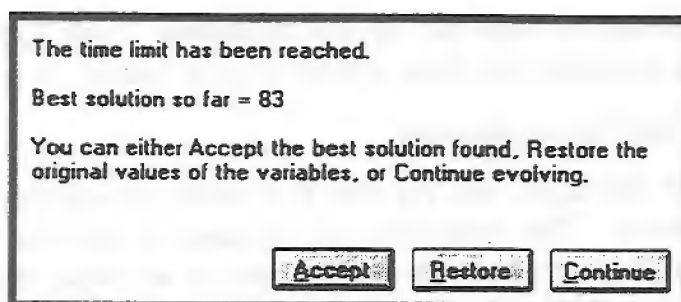
Once the problem is defined and the stopping conditions are set, you can run Evolver by clicking the OK button. Evolver will redraw the screen, showing

you how it is trying different scenarios. It is using the genetic algorithm to weed out the poor solutions and let the fittest solutions survive and reproduce. For more information about how Evolver is searching for solutions, see [Chapter 7: About Optimization](#).

Regardless of whether you have asked Evolver to redraw the screen, you can always tell if Evolver is running by the message in the status bar located at the bottom of your screen. The status bar also shows the best solution Evolver has found so far (see below).

Evolver in progress: press Esc to interrupt... Best=66

Evolver will stop after the 3 minutes have passed, and will present you with the following stop alert window:



The Stop alert tells you when your stopping conditions have been met.

The three options are:

Accept	Restore	Continue
This is the default selection. Accept will place all of the variable values from the best solution so far into your spreadsheet.	This option restores all of your variables to their original values, and returns to the spreadsheet as it was before you ran Evolver.	This option allows you to continue searching for solutions. When your original stopping condition/s are met again, Evolver will stop again.

23. Click on the "Accept" button.

You will be returned to the tutorial spreadsheet, with all of the new values that created the solution. Although in this example Evolver found a solution which yielded a total score of 83, your result may be higher or lower than this. Evolver may also find a different combination of workers to tasks that produced the same total score. These differences are due to the nature of Evolver's genetic algorithm engine, and they are the reason why Evolver is

Evolver User's Guide

able to solve a wider variety of problems, and find better solutions (see Chapter 7: About Optimization for more information).

If you would like to save the solution that Evolver produced, but would also like to retain a copy of the original spreadsheet, be sure to do a "save as" after you accept, to save the sheet under a new name. Then when you close or quit the original, do not save changes.

Running multiple programs

Evolver's genetic algorithm technique is very compute-intensive, and requires the Excel program to lock in a loop. However, you can break out of that lock and have Evolver work while you run other applications by following the steps below:

1. **At any time while Evolver is running, press and hold down the "Control" key while you hit the "Esc" key.**

This key combination will call up the Windows "Task List" window, or whatever task manager you have added to your system (e.g. METZ).

2. Select the Program Manager

From Program Manager, you are free to launch any application without disturbing Evolver. This technique can be used to run the Evolver Watcher standalone program (included with Evolver) to see what Evolver is doing.

Resetting Evolver

If you save the "tutorial.xls" spreadsheet, all of the Evolver settings will be saved along with the sheet. If you quit Excel and do not save your changes, you will get the original Evolver settings the next time you open this worksheet.

<p>NOTE: Any time you call Evolver while in Excel, all of the Evolver settings will be saved along with the spreadsheet that was open. The next time that sheet is opened, all of the most recent Evolver settings still be stored in Evolver.</p>

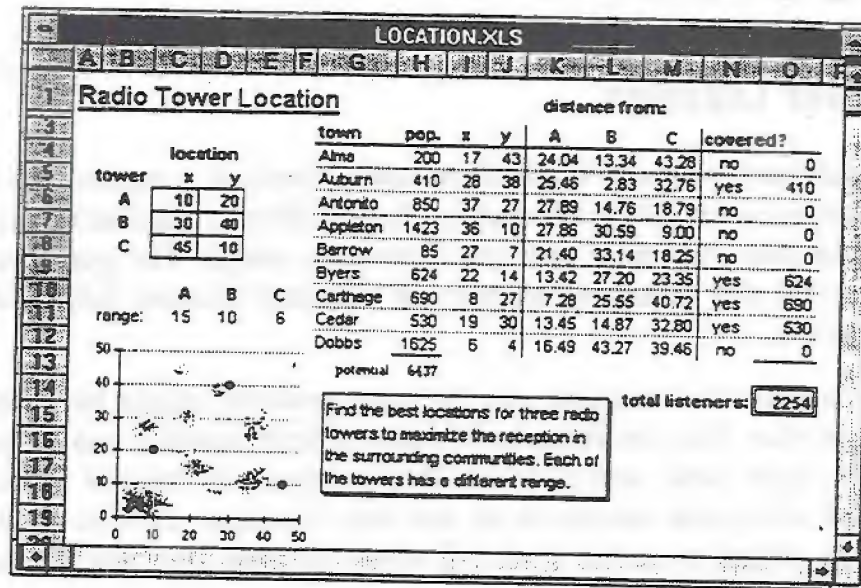
Radio Tower Location

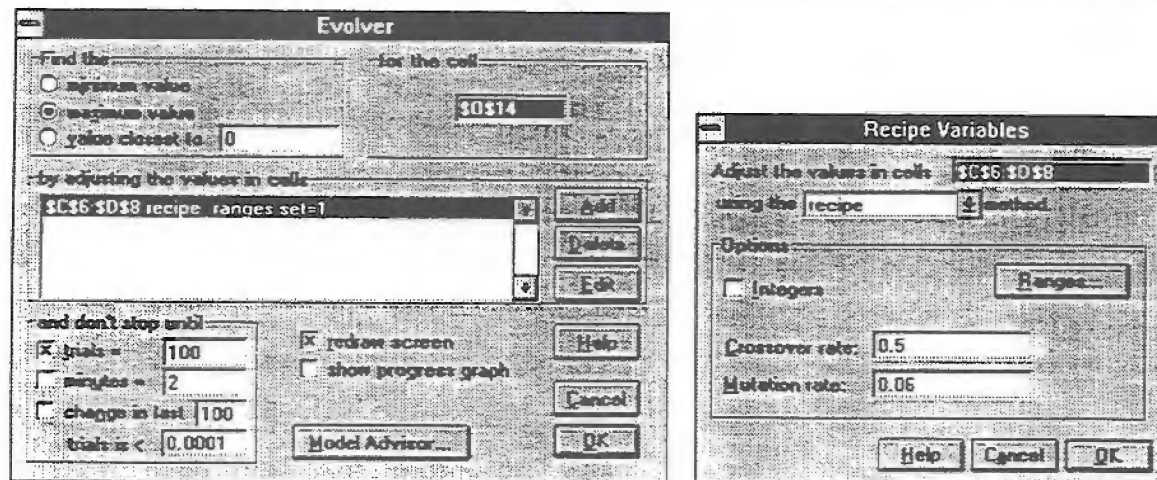
A radio network wants to build three radio towers in a region that has eight major communities. Each community has a different population size, and each radio tower has a different strength (broadcast range). The goal is to place the towers so that the maximum number of potential listeners fall inside the radii of the towers.

A more complicated example of a location problem might be to locate several factories so that they are a) in the vicinity of both vendors and customers, b) in affordable, open land, and c) near a large, technically trained work force. Any number of additional influences on the best locations, such as tax incentives, can also be added to such a model. Evolver can then find the best locations in x,y coordinate space.

If only one radio tower or one factory needed to be located, the problem would be relatively easy to solve; the factory could be moved about until all constraints were adequately met. If several or many objects need to be located and the location of one affects the location of another, then the problem is complex enough to warrant using Evolver.

Goal:	Find the best x,y coordinates for three radio towers so that the maximum potential listening population falls inside their broadcasting range.
Similar problems:	Find sites for warehouses that minimize the shipping necessary between warehouses and stores. Locate fire stations so that populations are best covered with a limited number of stations, including factors such as housing density.
Example file:	location.xls
Solving method:	recipe





The "recipe" solving method tells Evolver to adjust the variables chosen in any way it sees fit. As is the case with a recipe for baking, we are trying to find the right mix of "ingredients" (x,y coordinates) to produce the optimum solution.



The first of these is the fact that the system is not a simple one. It is a complex one, and it is one that is not easily understood. The second is the fact that the system is not a simple one. It is a complex one, and it is one that is not easily understood. The third is the fact that the system is not a simple one. It is a complex one, and it is one that is not easily understood.

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

-
- Always attempt the homework prior to the class it is due!
 - Homework is for class discussion only. It is not graded.
 - Never spend more than 45 minutes on the homework exercise!
 - If you can't understand it in 45 minutes, then bring your questions to class.
-

Homework 1: Lotek Industries (Getting Started)

If you have not used Excel 5.0 before, run appropriate examples and demos located under the help menu in Excel. Topics relevant to this exercise include: working in workbooks, selecting cells, using toolbars, entering data, creating formulas and links, editing a worksheet and formatting a worksheet. If you have used Excel 4.0 before, run the Quick Preview under the help menu to help acquaint you with the differences between version 4 and version 5.

You are the president of Lotek Industries, a consumer durables manufacturer. Your board chair and CEO, Megan Oldmoney, has asked you to look into an opportunity to acquire a plant that manufactures Koolbreez brand home air conditioners. The current owner, Rustbelt, Inc., is willing to sell the Koolbreez plant for \$2,000,000, and has faxed you a copy of their 5-year corporate planning model for the Koolbreez unit. The model, written in Excel version 5.0, is shown on the back of this sheet.

Type the model into Excel. Format the model identically as the one on the back of this sheet. There are three sections. The section labeled "Model Assumptions" contains constant parameters that are referred to elsewhere in the model.

The section labeled "Income Assumptions" contains formulas. Market growth increases 0.2% annually. Thus, cell C12 should contain the formula $=B12+.002$. Production cost per unit is 400 in 1995 then increases 1% per year (e.g., 1996 Production cost per unit $= B13 \times (1 + \$B\$7)$).

The Proforma Income Statement computes revenue, gross margin and income. Compute 1995 sales as the product of total market times initial market share. 1996 sales are the product of 1995 sales times $(1 + \text{market growth for 1995})$. Compute sales for subsequent years the same way. Price per unit is \$520 in 1995. Revenues equal price per unit times sales. Total fixed costs are \$2,850,000. Both price per unit and fixed costs increase by $(1 + \text{cost growth})$ in subsequent years (e.g., 1996 Price per unit $= B17 \times (1 + \$B\$6)$). Cost of goods sold equals production cost per unit times sales. Gross margin equals revenues minus cost of goods sold. And finally, income equals gross margin minus total fixed costs. All totals use the SUM() formula except cell G17. This value is the average price over the 5-year period. Use the AVERAGE() formula to compute this value.

1. What if initial market share is 5%? Is income greater than zero for each of the five years?
2. What is Koolbreez's total 5-year income if total market size was only 490,000?

	A	B	C	D	E	F	G	H
1	Lotek Industries							
2	Homework #1							
3	OPIM 101							
4								
5	Model Assumptions							
6	Cost growth	2.0%						
7	Production cost growth	1.0%						
8	Total market	500,000						
9	Initial market share	6.0%						
10								
11	Income Assumptions	1995	1996	1997	1998	1999		
12	Market growth	2.5%	2.7%	2.9%	3.1%	3.3%		
13	Production cost per unit	400.00	404.00	408.04	412.12	416.24		
14								
15	Proforma Income Statement	1995	1996	1997	1998	1999	Total	
16	Sales	30,000	30,750	31,580	32,496	33,503	158,330	
17	Price per unit	520.00	530.40	541.01	551.83	562.86	541.22	
18	Revenue	15,600,000	16,309,800	17,085,168	17,932,251	18,857,913	85,785,132	
19								
20	Cost of Goods Sold	12,000,000	12,423,000	12,886,005	13,392,296	13,945,532	64,646,834	
21	Gross Margin	3,600,000	3,886,800	4,199,163	4,539,954	4,912,381	21,138,298	
22								
23	Total Fixed Costs	2,850,000	2,907,000	2,965,140	3,024,443	3,084,932	14,831,514	
24	Income	750,000	979,800	1,234,023	1,515,511	1,827,450	6,306,784	
25								
26								

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 2: Lotek Industries (Model Revisions)

Lotek's strategic planning methodology is to analyze investment opportunities using net present value of income. Projects with a positive NPV are worth pursuing; projects with a negative NPV are not worth pursuing. Using the NPV function and a discount rate of 15%, find the NPV of the project for each year in the five-year period. Insert two lines into the model assumptions section. One for the discount rate of 15%; the other for the investment of \$2,000,000. Then, create named cells for each parameter in the model assumptions section. Once this is complete, the formula for B31 should be =NPV(Discount_rate,B29)-Investment; the formula for B32 should be =NPV(Discount_rate,B29:C29)-Investment, etc. Cell F31 should contain the NPV which encompasses the full five-year planning horizon.

1. Is Koolbreez a good deal from an NPV perspective? (Is cell F31 positive?)

To aid the usability of the model and interpretation of the results, use an IF statement in cell G31 to display whether Lotek should invest or not invest. The if statement has the form: =IF(logical_test, value if true, value if false). The logical test is a statement that evaluates to TRUE or FALSE (e.g., compare whether $F31 \geq 0$). The "value if true" could be a numeric value, text or a formula. The formula for cell G31 should be:

=IF(F31>0,"Invest","Do Not Invest")

You recently had lunch with Buzz Bowtie, Lotek's VP of Marketing, who told you about an error in your planning model. The market share data are based on an advertising campaign but the model does not contain advertising costs. Buzz estimates that Lotek's ad agency, Bucks and Morebucks, would charge an advertising fee of \$800,000 in 1995 and \$400,000 in each of the subsequent four years. Add a new line called "Advertising expense" to the income assumption section of the model.

2. Is the Koolbreez investment still a good deal from an NPV perspective? (Is cell F31 positive?)

To facilitate sensitivity analyses, add a new line called "Market share growth rate" to the model. Enter 0.2% as a constant in cell B10. Change the market growth line to reflect the parameter value for market share growth rate rather than a fixed number in each cell. These changes will help prevent errors in market share growth analysis.

	A	B	C	D	E	F	G	H	I
1	Lotek Industries								
2	Homework #2								
3	OPIM 101								
4									
5	Model Assumptions								Reference
6	Cost growth	2.0%							=Homework2!\$B\$6
7	Production cost growth	1.0%							=Homework2!\$B\$11
8	Total market	500,000							=Homework2!\$B\$9
9	Initial market share	6.0%							=Homework2!\$B\$12
10	Market share growth rate	0.2%							=Homework2!\$B\$10
11	Discount rate	15.0%							=Homework2!\$B\$7
12	Investment	2,000,000							=Homework2!\$B\$8
13									
14	Income Assumptions	1995	1996	1997	1998	1999			
15	Market growth	2.5%	2.7%	2.9%	3.1%	3.3%			
16	Fixed costs	2,850,000	2,907,000	2,965,140	3,024,443	3,084,932			
17	Advertising expense	800,000	400,000	400,000	400,000	400,000			
18	Production cost per unit	400.00	404.00	408.04	412.12	416.24			
19									
20	Proforma Income Statement	1995	1996	1997	1998	1999	Total		
21	Sales	30,000	30,750	31,580	32,496	33,503	158,330		
22	Price per unit	520.00	530.40	541.01	551.83	562.86	541.22		
23	Revenue	15,600,000	16,309,800	17,085,168	17,932,251	18,857,913	85,785,132		
24									
25	Cost of Goods Sold	12,000,000	12,423,000	12,886,005	13,392,296	13,945,532	64,646,834		
26	Gross Margin	3,600,000	3,886,800	4,199,163	4,539,954	4,912,381	21,138,298		
27									
28	Total Fixed Costs	3,650,000	3,307,000	3,365,140	3,424,443	3,484,932	17,231,514		
29	Income	-50,000	579,800	834,023	1,115,511	1,427,450	3,906,784		
30									
31	Net Present Value	-2,043,478	-1,605,066	-1,056,683	-418,885	290,809	Invest		

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

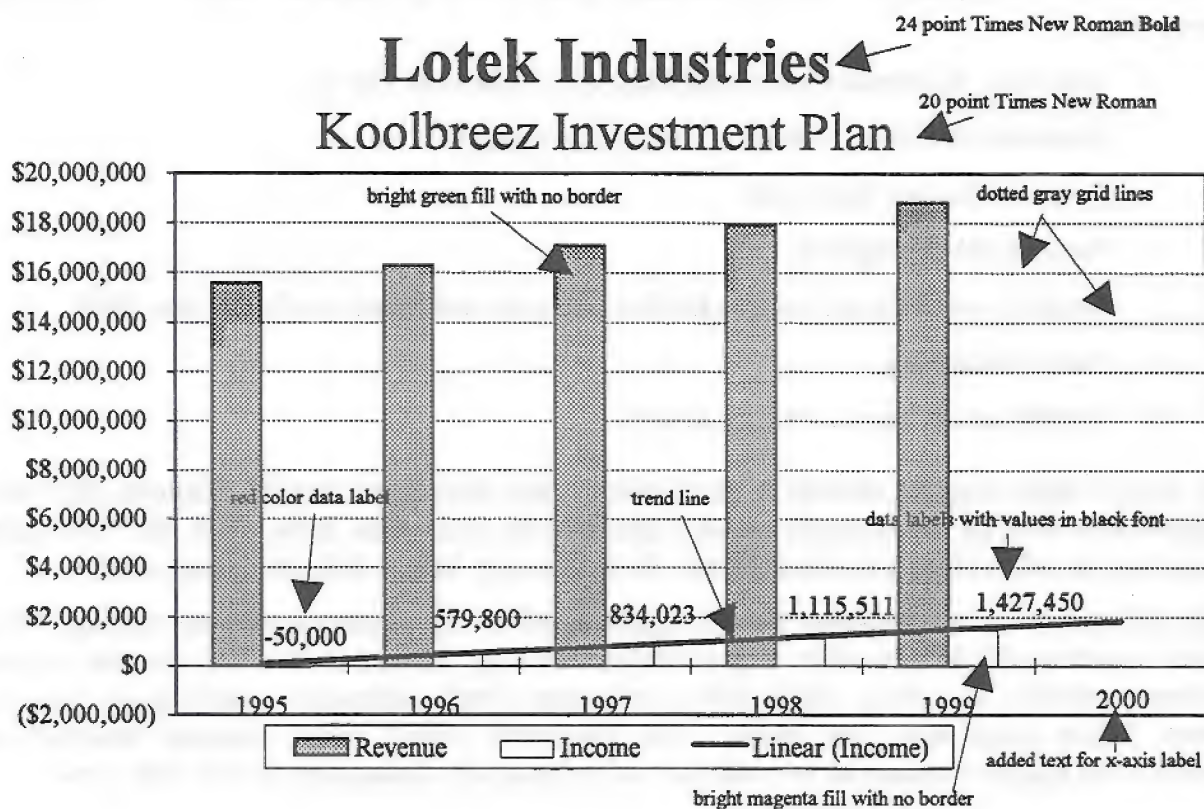
Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 3: Lotek Industries (Graphs)

Megan Oldmoney, CEO and Chair of the board, has asked for a graph of revenues and income for the five-year Koolbreez investment plan. She would also like a projection of income in the year 2000. Prepare a graph like the one shown below. The added labels show color or added features.

You begin the task by brushing up on Excel 5.0 graphing features. Examine the topics under the examples and demos help menu item in Excel. Three menu topics contain help for graphing data. "Creating a chart" contains a basic introduction to graphing. "Formatting a chart" explains how to change scale information. "Using charts to analyze data" shows how to add a trend line to your graph. You can also use the Workshop III section of the Excel 5 Superbook as a reference guide. Page 41 contains help on selecting noncontiguous ranges. Once the three rows containing dates, revenues, and income are selected, use the Insert Chart submenu to insert a new sheet in your workbook and follow the Chart Wizard prompts to create your graph.



Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 4: Lotek Industries (Slide Show)

Megan Oldmoney, CEO and Chair of the board, was impressed with the graph and has asked you to make a slide presentation to the board of directors. She would like you to highlight the current situation with the Koolbreez investment decision and state your recommendation.

You begin the task by brushing up on Powerpoint 4.0 slide show features. Quick preview on the Powerpoint 5 help menu is helpful for getting started. Create a new slide show using a template wizard created by an Powerpoint macro. Depending on how your machine is configured Powerpoint will open a dialog box for creating a new presentation (or select File New from the menu bar). Use the Auto Content Wizard and fill in boxes as prompted. Press next to continue from one screen to the next within the Auto Content Wizard. When Auto Content Wizard is finished, Cue Cards provide tips for working with Powerpoint. Switch to slide view to edit the content supplied by the Auto Content Wizard. Replace graphs and text with material for the Homework #4 presentation. Your preliminary notes for the presentation contain the following seven slides.

- ◇ Title slide Koolbreez Investment Plan note: add some clip art
- ◇ Statement of Today's situation - should we invest?
- ◇ Proforma Income Statement
- ◇ State model assumptions
- ◇ Graph of revenues and income for five-year plan with trend line for the year 2000
- ◇ Recommendation
- ◇ Identify action items - what to do next

To create a slide, copy the desired range of cells or chart from Excel into the Windows clipboard. Select Paste to copy the contents of the Clipboard into your slide show. Use the Powerpoint menu bars to select video transition effects. Each slide may have a different special effect.

The real issue is one of designing visually appealing slides that communicate your message. Use borders, pattern fill to add color to your slides. Change the font colors and sizes to capture audience attention as well as add legibility for greater viewing distances. Add clip art from the Insert Object menu into your slides. The Microsoft Clipart gallery contains hundreds of professional-quality commercial artwork that will enhance the appearance of your slide show.

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

Always attempt the homework prior to the class it is due!

Homework is for class discussion only. It is not graded.

Never spend more than 45 minutes on the homework exercise!

If you can't understand it in 45 minutes, then bring your questions to class.

Homework 5: Lotek Industries (Sensitivity Analysis)

Buzz Bowtie, Lotek's VP of Marketing discussed the Koolbreez advertising plan with Lotek's ad agency, Bucks and Morebucks. Buzz estimates that a stronger ad campaign could increase market share for Koolbreez from its current level of 6% to 7% beginning in 1995. This would require advertising fees of \$1,600,000 in 1995 and \$800,000 in each of the subsequent four years. Analyze and report the net present value of this alternative advertising campaign.

The total market of 500,000 units is Rustbelt's estimate. Since Rustbelt Inc. is selling the Koolbreez plant, you are leery of their estimate. You consult with George Guesswell, Forecast Analyst with Lotek's Corporate Planning Staff. George indicates that although the 500,000 estimate is pretty accurate, depending on economic conditions, it conceivably could vary from 490,000 to 515,000. Analyze and report the NPV of two alternative scenarios: a best case total market of 515,000 and a worst case total market of 490,000. (Assume that this is the only change to the base model).

Megan Oldmoney stops by your office on her way home at 6:30 in the evening: "I'd like to present our Koolbreez proposal to the Board at 9AM tomorrow, but first I wanted to find out what you think about the deal? Also, what is our IRR for the project?" The IRR, internal rate of return, is the discount rate when the NPV is zero. You know this will be easy to compute using the goal seek command in Excel, but how should you respond to Oldmoney's other concerns? Should Lotek invest in the Koolbreez deal or not? If so, do you recommend the ad campaign? Which total market value should you use? best case? worst case? both? neither?

While presenting your recommendations on Koolbreez to the board of directors, board member J. Parker Pinstripe III interrupts you in mid-sentence: "It seems to me your analysis assumes that unit production costs will only increase at 1% per year. I happen to play golf with the CEO of a key supplier for Koolbreez who tells me his company is currently negotiating its labor contract, which will probably require them to raise prices. You might want to fold this issue into your thinking before we commit on this acquisition." Walking out of the meeting, Oldmoney casually mentions to you: "Let me know if you get a chance to look at the impact of that supplier on possible unit production cost increases. I'm having lunch over at the Country Club with the boys from Rustbelt today, and they're anxious to finalize this Koolbreez deal. If I don't hear from you before lunch, I'll assume production costs are a non-issue."

"What if" overload has given you a mega-headache. Your stomach is churning and the TUMS bottle is empty. How can you manage all these model assumptions and make sense of all these analyses? You call George Guesswell but he is not in. His secretary refers you to Molly Modelwright a DSS wizard from the Wharton School. "No problem," says Modelwright, "just use the Scenario Manager in Excel. Each scenario can be labeled with a unique name like Advertising, Worst total market, Best total market, Production cost, base case, etc. Excel prompts you for cells that change in each scenario. Once entered you can quickly view each scenario with the Show button or display a summary report showing the effect of changed cells on a range of result cells like NPV. Chapter 39 in the Excel 5 Superbook has a good introduction to the scenario manager." You think what a relief, thank Molly and begin to examine the Lotek investment with new vigor.

Do you still plan to go ahead with the Koolbreez proposal? Why or why not?

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 6: Lotek Industries (Data Tables)

Your sensitivity analysis of the five-year Koolbreez investment plan is still not complete. You are now interested in examining the affects of discount rate and initial market share on net present value. The problem with scenario manager is that you can only see the result of one modification at a time. To study the effect a range of values has on a formula and view all numbers simultaneously, you need to use a data table. You plan to use a two-way data table (see pages 446-447 in the Excel 5.0 Superbook). The data table should be on the same worksheet as the model. You will vary the discount rate from 8% to 20% and vary the initial market share from 1% to 10%. Both ranges will be incremented by 1%. The formula is entered in the upper-left hand corner of the data table. Format your data table like the one shown below.

Complete the data table and determine the relationship between discount rate and initial market share on net present value. Does the data table provide any new critical insights regarding the Koolbreez plant investment decision?

Initial market share											
=NPV	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	
8%											
9%											
10%											
11%											
12%											
13%											
14%											
15%											
16%											
17%											
18%											
19%											
20%											

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 7: Lotek Industries (Monte Carlo Simulation and Risk Analysis)

Molly Modelwright, the DSS wizard from the Corporate Planning Staff, calls just as you finish the data table. She asks how the sensitivity analyses are going. You start griping that there are just too many possible combinations of input values to calculate every possible answer and ask her if there is a better way to figure out which model parameters are the most important. Molly said, "Monte Carlo simulation is an efficient way to handle these types of problems. Instead of calculating all possible combinations of input values, Monte Carlo simulation uses a random number generator to select a range of possible input values for a model. The result is expressed as a range of possible values using descriptive statistics. And besides, Monte Carlo simulation is easy to do using the Crystal Ball add-in under the Excel tool menu." "This is great," you tell her, "How do I get started?" She tells you about two tutorials (see Crystal Ball version 3.0 User Manual - handout #12 in the course pack).

The first tutorial, "Futura Apartments" simulates profit/loss projections from apartment rentals. This tutorial is ready to run so you can quickly see how Crystal Ball works. If you work with statistics and forecasting techniques, this may be all the introduction you need before running your own models with Crystal Ball. The second tutorial, "Vision Research", gives you a chance to enter data and set up a complete simulation for a major corporate expenditure decision. As you work through the second tutorial, do not worry about making mistakes; recovery is as easy as backing up and repeating the steps.

After completing the tutorials you have learned a lot about building models for making decisions under uncertainty. Now it is time to incorporate these new skills in the Koolbreez plant investment decision. For the model parameter assumptions, use the uniform distributions in the table. Use NPV as the forecast cell. Run the model for 1000 iterations. and create a report of the results. There are two big questions to address. First, how certain are you of attaining an NPV greater than zero? Second, what model parameters explain the most variance in NPV?

Model Parameter	Minimum	Maximum
Cost growth	0.0%	4.0%
Production cost growth	0.0%	3.0%
Total market	450,000	550,000
Initial market share	4.0%	8.0%
Market share growth rate	0.0%	.5%
Discount rate	8.0%	20.0%
Investment	\$1,500,000	\$2,000,000

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

Always attempt the homework prior to the class it is due!
 Homework is for class discussion only. It is not graded.
 Never spend more than 45 minutes on the homework exercise!
 If you can't understand it in 45 minutes, then bring your questions to class.

Homework 8: Big Mac Attack

1. Download the template homewk8.xls from the student network. It should be identical to the one on the following page. Save the file for homework 9-11.

2. Study the rows and columns of the worksheet.
 Column A contains a list of items that can be ordered from McDonald's.
 Column B contains the price of each item. (Note that these prices may not reflect the actual price at your favorite McDonald's, but at the time, these were the actual prices.)
 Column C contains the total calories for each item.
 Column D through column M contain data amount the nutrients and vitamins contained in each food item.
 Column N contains a quantity for each food item. (Note: all cells except N5:N19 are locked and password protected. Thus, these are the only values on the spreadsheet that may be altered.)

 Row 21 contains the column totals. The TOTAL row is computed by using the SUMPRODUCT function of Excel which computes the dot product of two vectors. For example, total cost is the product of quantity times price.
 B21 is the total cost, \$7.81, associated with purchasing 3 Hamburgers, 4 boxes of Wheaties, and 3 Milks.
 C21 is the total calories, 1455, for 3 Hamburgers, 4 boxes of Wheaties, and 3 Milks.
 D21 through M21 represent the total amount of protein, fat, sodium, vitamin A, C, B1, B2, niacin, calcium, and iron in that diet.
 Row 22 represents minimum USDA requirements for each of these essential food groups or vitamins.
 Row 23 represents maximum USDA requirements for each of these essential food groups or vitamins. What does the formula in E23 represent? There are 9 cal/gram of fat.

3. Suppose you decide to purchase all of your food for one day from McDonald's. You are extremely concerned with eating a proper diet the meets or exceeds the USDA recommended guidelines. However, you only have \$20 cash until your next check and you would like to have enough money left-over to go out later that night. Change the values of cells N5:N19 to find the amount of each item on the menu that will minimize cost and still meet all of the USDA requirements?

A B C D E F G H I J K L M N

¹ Big Mac Attack

2

3	4 McDonald's Menu Item	Price \$	Calories	Protein grams	Fat g.	Sodium mg.	A	C	B1	B2	Niacin	Calcium	Iron	Diet
							Percent of USDA requirements							
5	Hamburger	0.59	255	12	9	490	4	4	20	10	20	10	15	3
6	McLean Deluxe	1.79	320	22	10	670	10	10	25	20	35	15	20	0
7	Big Mac	1.65	500	25	26	890	6	2	30	25	35	25	20	0
8	Small Fries	0.68	220	3	12	110	0	15	10	0	10	0	2	0
9	Chicken	1.56	270	20	15	580	0	0	8	8	40	0	6	0
10	Honey	0.00	45	0	0	0	0	0	0	0	0	0	0	0
11	Chef Salad	2.69	170	17	9	400	100	35	20	15	20	15	8	0
12	Garden Salad	1.96	50	4	2	70	90	35	6	6	2	4	8	0
13	Egg McMuffin	1.36	280	18	11	710	10	0	30	20	20	25	15	0
14	Wheaties	1.09	90	2	1	220	20	20	20	20	20	2	20	4
15	Vanilla Yogurt	0.63	105	4	1	80	2	0	2	10	2	10	0	0
16	Milk	0.56	110	9	2	130	10	4	8	30	0	30	0	3
17	Orange Juice	0.88	80	1	0	0	0	120	10	0	0	0	0	0
18	Grapefruit Juice	0.68	80	1	0	0	0	100	4	2	2	0	0	0
19	Apple Juice	0.68	90	0	0	5	0	2	2	0	0	0	4	0
20														
21	TOTAL	7.81	1455	71.0	37.0	2740	122	104	164	200	140	128	125	
22	at least			55			100	100	100	100	100	100	100	
23	at most				48.5	3000								

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

Always attempt the homework prior to the class it is due!
 Homework is for class discussion only. It is not graded.
 Never spend more than 45 minutes on the homework exercise!
 If you can't understand it in 45 minutes, then bring your questions to class.

Homework 9: Big Mac Attack LP

Part 1. Homework 8 used a trial and error technique to find the amount of each item on the menu that minimized total cost and still met all of the USDA requirements. Use Excel Solver to determine whether you found the correct answer. Solver is on the Tools menu.

Set target cell \$B\$21 equal to Min by changing cells \$N\$5:\$N\$19

Subject to the structural constraints that the diet has:

- greater than or equal to the minimum grams of protein
- less than or equal to the maximum grams of fat
- less than or equal to the maximum grams of sodium
- greater than or equal to the minimum percent of vitamin A
- greater than or equal to the minimum percent of vitamin C
- greater than or equal to the minimum percent of vitamin B1
- greater than or equal to the minimum percent of vitamin B2
- greater than or equal to the minimum percent of niacin
- greater than or equal to the minimum percent of calcium
- greater than or equal to the minimum percent of iron

and the non-negativity constraints that the amount of the food items in N5:N19 must be greater than or equal to 0.

Clicking on "Solve", after entering the constraints of the problem produces a solution and prompts whether you would like to generate reports.

Questions: What is the minimum cost? How much of each food item should be purchased? Is the solution realistic?

Part 2. Part 1 asked to find a diet that minimizes cost. In part 2, find the diet that maximizes cost. The maximum cost diet is relevant in determining the cost range of all feasible diets as well as for studying palatability, variety maximization, and compromise diets. If the range is wide, then there is a large number of feasible diets. If the range is narrow, then there is a small number of feasible diets.

Questions: What is the maximum cost? How much of each food item should be purchased? Is the solution realistic? How much more money is this diet compared to cost minimization?

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 10: Big Mac Attack - Sensitivity Analysis

Part 1. Homework 9 used the Excel Solver to determine the optimum solution to the cost minimization problem. Resolve the cost minimization model. Prior to clicking on "Solve", click on Options. Under the Options box click on "assume linear model". After entering all the components to the optimization problem to produce a solution to your model, click on "Solve". Excel displays a dialogue box labeled

"Solver Results
 Solver found a solution. All constraints
 and optimality conditions are satisfied.

Excel also allows you to create Answer, Sensitivity, and Limits reports. Hold down the Ctrl key and use the mouse to select the Answer and Sensitivity reports. These two reports are added as new worksheets to your workbook. If your Sensitivity report does not contain "Allowable Increase and Allowable Decrease" for Objective Coefficient and Shadow Price, then you forgot to click on "assume linear model" under the options box.

Part 2. Use the information in the Answer Report to answer the following questions:

1. If a constraint is binding, how much slack (unused resource) is available? Zero
2. How many decision variables (adjustable cells) does the model contain? 15
3. How many binding constraints does the model contain? 15
4. Is there any relationship between the number of decision variables and the number of binding constraints? number of binding constraints \geq number decision variables
5. Are all the binding constraints structural? 5 structural; 10 non-negativity constraints
6. Suppose McDonald's is out of orange juice. How much would you be willing to pay for one orange juice from another source to help you meet the cost minimization diet? Assume it is the same size and quality. Orange Juice is not binding, therefore \$0.00

Part 3. Use the information in the Sensitivity Report to answer the following questions:

Shadow Prices (Convert output so that shadow price has 5 decimal places)

1. Why is the shadow price for Total grams protein 0.00000? Non-binding constraints do not have a shadow price.
2. Do all the binding constraints have a shadow price? Yes
3. If the USDA adopted a 50% increase in the vitamin C requirement, what is the increased cost to the cost-minimizing diet associated with this change?
 $\$.007 / \text{percent change in vitamin C} \times 50\% = \$.35 \text{ increase}$

Reduced Costs (Convert output so that reduced cost has 5 decimal places)

4. McDonald's adopted a policy of charging \$0.10 for honey. What is the affect of this policy on the cost-minimizing diet? cost minimizing diet does not use honey, no effect
5. How much would the price of orange juice have to decrease, before it would be attractive to consider adding orange juice to the cost-minimizing diet? by the amount of its reduced cost, \$.064
6. If you were forced to include one Egg McMuffin in your diet (and adjusted other items in your diet to reflect the nutritional value of an Egg McMuffin), by how much would the cost of the cost-minimizing diet increase? by the amount of its reduced cost, \$.561

Right-hand side ranging (Allowable increase and Allowable decrease)

7. Suppose you are at risk to coronary problems and your Doctor limits your daily fat to 40 grams. Is the cost-minimizing diet still acceptable? Why or why not? No, the cost-minimizing diet allows 52.5 grams of fat. 40 grams is below this value and outside the RHS lower bound ranging limit.
8. Suppose your Doctor puts you on a low-sodium diet and decreases your daily sodium limit to 2200 mg. Is the cost-minimizing diet still acceptable? Why or why not? Sodium is a binding constraint with the cost-minimizing diet. 3000 mg sodium will be consumed; therefore, the diet must change. Shadow prices allow us to determine that the new cost-minimizing diet will cost \$6.36 because 2000 mg sodium is within the RHS ranging limits. $-0.000298 \times 800 = \$.238$ $\$6.126 + .238 = \6.364

Objective ranging (Allowable increase and Allowable decrease)

9. McDonald's increases the price of hamburgers by \$0.08, the price of garden salads by \$0.20, the price of Wheaties by \$0.25, and the price of milk by \$0.30. Is the cost-minimizing diet still acceptable? What is the cost of the cost-minimizing diet under the new pricing policy? Amount of each of the 15 food items does not change. Cost will increase by $\$.08 \times 5.299 + \$.20 \times .535 + \$.25 \times .812 + \$.30 \times 1.441 = \$1.166$ (\$7.29)
10. Suppose the price of garden salad increased by \$0.50. What affect does this pricing policy have on the optimal solution for the cost-minimizing diet? Is the cost-minimizing diet still acceptable? A 50 cent price increase is outside the upper bound for objective ranging, therefore the effect on cost can not be determined without resolving. However, the diet composition would change.

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 11: Big Mac Attack Integer Programming

Part 1. Homework 9 and 10 allowed the amount of each item on the menu to be floating point or fractional values. Thus, the model is very unrealistic since it is difficult to purchase 5.3 hamburgers. Add the constraint that the amount of the food items in N5:N19 must be integer values. **N5:N19 Int Integer**

Solve the problem with the new constraints. Questions: What is the minimum cost? How much of each food item should be purchased?

Part 2. Instead of minimizing cost, suppose we are on a diet and need to minimize calories. Questions: What is the new minimum cost for this diet? How much of each food item should be purchased? How much more expensive is the calorie-minimizing diet compared to the cost-minimization diet? What percent fewer calories does the new diet have compared with the cost-minimization diet?

Part 3. Return the model to the cost minimization model in Part 1 above. The cost-minimizing diet is boring and the cost-maximizing diet is too expensive as well as boring (Salad, yogurt, and Grapefruit juice). Suppose you create a compromise diet subject to all the constraints in Part 1 above plus the following constraints:

At least one Egg McMuffin or Wheaties
One order of Fries with each burger ($N5+N6+N7$)
At least three drinks (milk or juice)
No more than two Wheaties
Number milk equal number Wheaties
In addition to any milk taken with Wheaties at least two more drinks (milk or juice)
At least one salad (Chef or Garden salad)
No more packages of honey than Chicken

Solve the problem with the new constraints. Questions: What is the minimum cost? How much of each food item should be purchased? Is the variety more palatable?

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 12: Braincel Tutorial

BCDATA.xls contains 17 records of data on loan applications as well as a loan officer's indication of the applicant's ability to repay the loan on a scale of 1 to 5. A "1" means very poor loan repayment probability and "5" means excellent loan repayment probability. Other fields for each applicant include: monthly income, monthly expense, owns home, years at present job, years at previous job, years at present address, years at previous address, and number of dependents. Of course, a loan neural network would need a larger database than 17 records. However, the tutorial demonstrates how to use Braincel to make loan repayment forecasts based on previous applications.

Enable Braincel by selecting the file braincel.xll from the braincel directory. Complete the Braincel tutorial on pages 158-165 of the course pack (Part A).

Homework 13: Evolver Tutorial

Tutorial.xls contains a spreadsheet describing a resource allocation problem where each worker must perform one task. The table rates how well each of ten workers performs each of ten tasks. Each worker's ability to perform each task is rated on a scale of 0 to 10 (0=can not do the task, 10=perfect at the task). The optimization problem is to match each worker to a task so that overall productivity is maximized. The "Chosen Task" (column N) to the right of ratings assign each worker arbitrarily to one task. The next column (column P) enters each worker's rating for that task. Finally, the "Total Score" is the sum of the individual ratings for all 10 tasks.

There are $10!$ ($10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 3,628,800$) possible ways to assign workers to their tasks. Although this problem may seem simple, the problem's complexity increases exponentially as we add additional constraints (e.g., a certain task requires two workers).

Using Evolver to solve the problem is similar to using Solver. Evolver is on the Tools menu. Load Evolver and solve this problem by following the instructions in the tutorial on pages 173-184 of the course pack (Part A).

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

-
- Always attempt the homework prior to the class it is due!
 - Homework is for class discussion only. It is not graded.
 - Never spend more than 45 minutes on the homework exercise!
 - If you can't understand it in 45 minutes, then bring your questions to class.

Homework 14: Database Forms and Complex Queries Using Data Tables

This exercise provides practice working with Excel 5.0 commands used to manipulate databases or lists. Download the template homewk14.xls from the student network. This file contains the OPIM 101 class list (except for names and social security numbers). This list contains the field names: Order, Section, College, Major, and ID Code. Each field is a column and the columns contain field values (e.g., the field "Section" has values of 1, 2, or 3). Each record in the list is a row that contains a unique collection of field values. This list does not contain duplicate records.

Data Forms are a dialog box used to add, delete, edit, and find records in a list. Copy the field names to cells A13:F13 and define the named range **Criteria** as \$A\$13:\$F\$14 and **Database** as \$A\$23:\$F\$290. Select form under the Data menu to open the data form dialog box. *Add* a new record for student 300.

Order	Section	College	Class	Major	ID Code
300	4	WH	SR	UNDC	XYZ

After adding the record, examine the named range "Database". Excel automatically adjusts the list range to include new records in the named range. It also prevents you from overwriting existing records on your spreadsheet.

Now *delete* the record you just added. Again note that Excel automatically adjusts the list range to delete the record from the named range "Database".

Use the criteria feature to find out how many marketing students are enrolled in OPIM 101. Select the criteria button then enter MKTG in the field for majors. The data form dialog box displays the first record found. The Find Prev and Find Next buttons can be used to navigate through the selected records.

Find number of marketing students enrolled in OPIM 101. ☐

Use the criteria fields to complete a more complex search. Find number of students who are juniors in the College with a declared major (not undecided <> UNDC) enrolled in OPIM 101.

☐

Find all freshmen with majors declared as undecided and all Wharton seniors. ☐

The criteria range in the data form dialog box can not handle complex queries. Use database functions and named criteria ranges for more complex queries. "Find all freshmen with majors declared as undecided and all Wharton seniors" is a compound query. To specify multiple criteria for a single query use multiple lines in the criteria range. One line specifies freshmen (FR) undecided (UNDC); the other specifies Wharton (WH) senior (SR). Copy the field names to the cell range A2:F2 and complete the query. Use the database function =DCOUNT(Database,"Order",A2:F4) to find how many students in OPIM 101 fit these criteria? ☐

Change the criteria range to C2:D4. What does this value represent? How many students in OPIM 101 fit these criteria? ☐

In addition to multiple lines, criteria ranges may contain multiple copies of the field names. Suppose you wanted to count all the records in the OPIM 101 class list with order value greater than 100 but less than or equal to 200. Copy the field names to the cell range A8:F8 and repeat the field name Order in cell G8. Use the values >100 and <=200 under the field names "Order" and the database function =DCOUNT(Database,"Order",A2:F4) to count the records that fit these criteria. Search for the topic "and/or criteria range" using Excel help to find additional examples.

Data Tables Use a data table to find the number of freshmen, sophomores, juniors and seniors enrolled in OPIM 101 from each college. The named ranges "Database" and "Criteria" will make it easier to build a Data Table using database functions. Enter the data table below the named range Criteria as shown below. Cell A15 contains the database function formula =DCOUNT(Database,"Order",Criteria) where "Order" is the field name of the values to be counted. Use the help feature in Excel to see examples of other database functions (e.g., DAVERAGE, DMIN, DSUM).

	A	B	C	D	E	F
13	Order	Section	College	Class	Major	ID Code
14						
15		FR	SO	JR	SR	
16	CGS					
17	COL					
18	EAS					
19	EVE					
20	WH					
21						

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

Always attempt the homework prior to the class it is due!

Homework is for class discussion only. It is not graded.

Never spend more than 45 minutes on the homework exercise!

If you can't understand it in 45 minutes, then bring your questions to class.

Homework 15: Database Sorting, AutoFilter, Subtotals, and Pivot Tables

This exercise provides practice working with Excel 5.0 commands used to manipulate databases or lists. Copy the OPIM 101 class list from homework 14 to a new worksheet.

Sorting a list facilitates viewing groups of records in the list. Sort the records in the OPIM 101 class list by College, Class, and Major. Record 11 should be the first row in the list after sorting.

Using the sorted list, obtain **subtotals** by College and Class using the Subtotal Data command. Subtotals must be executed twice. Select subtotals for College first. Then select subtotals for class. Be certain the box "Replace Current Subtotals" is not selected otherwise Excel will remove the previous subtotal for College. Click on the subtotal box 3 to collapse the record detail and show summary data for each group. Do these subtotals agree with the values in the Data Table from Homework #14?

Remove all subtotals by selecting the "Remove All" box in the subtotal dialog box. Select a cell in the OPIM 101 class list. Select the Data | Filter | **AutoFilter** command. Excel adds drop-down arrows to the cells containing the field names. Clicking on the "Major" field displays all the majors for all students in the class. By selecting "MKTG", only records of students with a marketing major will be displayed. How many records meet this criteria. Does this value agree with the value found in homework 14?

AutoFilter allows more complex filtering of record criteria by using multiple fields with custom sorts. Use the criteria fields to complete a more complex search. Find number of students who are juniors in the College with a declared major (not undecided <> UNDC) enrolled in OPIM 101. Does this value agree with the value found in homework 14? ☐

Define a criteria range and use the Data | Filter | **AdvancedFilter** command to find all freshmen with majors declared as undecided and all Wharton seniors. Does this value agree with the value found in homework 14? ☐

Use the **Pivot Table Wizard** to create a Pivot Table like the one below. Use **Section** as a page value, **College** as a row value, **Class** as a column value and count of **ID Code** as a data value. By selecting different page views, the Pivot Table displays the number of freshmen, sophomores, juniors and seniors enrolled in OPIM 101 from each college for each section or for all sections combined.

Section	1
---------	---

Count of ID Code	Class				
College	FR	JR	SO	SR	Grand Total
CGS	1	0	0	0	1
COL	3	3	2	4	12
EAS	0	1	1	0	2
EVE	1	0	0	0	1
WH	23	5	4	6	38
Grand Total	28	9	7	10	54

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

Always attempt the homework prior to the class it is due!

Homework is for class discussion only. It is not graded.

Never spend more than 45 minutes on the homework exercise!

If you can't understand it in 45 minutes, then bring your questions to class.

Homework 16: Visual Basic object properties, ranges, input boxes, message boxes

This exercise provides practice working with Excel 5.0 Visual Basic. The Visual Basic program will be written on the module sheet, named homework16; and a worksheet, named HW16, will be used to display portions of the program output. Write a Visual Basic program to prompt you for your first name and last name. Next, a message box asks whether you have entered these fields correctly. Finally, output and format the values in cells A1:A4.

First, familiarize yourself with the syntax for the commands used in this program by reading the following pages of the *Excel 5.0 Superbook*. [InputBox (pp. 723-725), MsgBox (pp. 720-723), Range().formula (p. 710), as well as commands for selecting cells (chapter 53) and recording macros (pp. 653-658)] Often many of the actions in a macro can be recorded using the Record Macro feature. Once recorded, these actions can be copied from one module sheet to another.

Begin the module with the statement Option Explicit (p. 683). This Visual Basic command will cause Visual Basic to generate an error message if variables are not declared prior to their use in the execution of the program. This is very useful for writing good code.

The elementary unit of Visual Basic programming is called a procedure. [Note: the terms macro and procedure are equivalent.] Begin your program by defining a sub procedure call homework16. Declare the variables lastname, firstname, fullname as String and answer and i as Integer.

Use two InputBox commands to prompt the user for their first and last name. The InputBox command creates a dialog box to prompt the user for input. The syntax for this method is: `firstname = InputBox(prompt:="Enter your first name.", Title:="Input Box")` where `firstname` is a variable declared as a String to accept the input from the InputBox command.

The concatenation operator & can be used to join the first and last name into the variable `fullname`. Insert appropriate spaces and a question mark to use the concatenated name in the message box prompt that asks the user whether the first and last names were entered correctly. The syntax for this method is: `answer = MsgBox("Is your name " & fullname, 3, "Verify Name")` The 3 is used to obtain a Yes No Cancel type of message box.

Use the record macro feature to record the Visual Basic commands to select a worksheet named HW16 and clear the contents of the entire sheet. Copy these commands into your homework 16 module. Next use the Range("A1").formula properties to assign the contents of the variables `firstname`, `lastname`, `fullname` to cells A1, A2, and A3 and put the answer the message box question in cell A4. Finally, use the record macro feature to record the Visual Basic commands to autofit the column width and make A1 the active cell on the worksheet HW16. Copy these commands into your homework 16 module.

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 17: Visual Basic control statements

This exercise provides practice working with Excel 5.0 Visual Basic. The Visual Basic program will be written on a module sheet, named homework17 and a worksheet, named HW17 will be used to display portions of the program output. The program computes the total electricity cost given kilowatt hour usage (kwh). Rates for electricity are shown in the table below. Use nested if then else statements or the select case statement to assign the appropriate rate. Use an InputBox to prompt the user for total kilowatt hour usage. Compute the total electricity cost. Display the total kilowatt hour usage, rate and total electricity cost in three successive rows of column A. Use a message box to ask the user whether to continue the calculations. If so, use a GoTo statement to transfer control back to the beginning of the input section of the program. Continue to display the total kilowatt hour usage, rate and total electricity cost in three successive rows of column A with a blank row between each output area. Do not erase previously displayed values.

First, familiarize yourself with the syntax for the new commands used in this program by reading the following pages of the *Excel 5.0 Superbook*. [If statements (pp. 689-693), select case (pp. 693-694)] Select case is much easier to use to solve this problem, but you may use either one. Begin the module with the statement Option Explicit (p. 683). Define a sub procedure called homework17. Declare the variables kwh, rate as Single; answer, i as Integer. Use i as a counter for rows. Initialize the value of i to 0. Create a line label for transferring control using a GoTo statement. Line labels are not case sensitive, but they must begin with a letter and end with a colon. Each line label must be uniquely named. Use an InputBox command to prompt the user for total kilowatt hour usage. Assign this value to the variable kwh. Next, create the nested if then else statements or the select case statements to assign the appropriate electric rate.

Use the record macro feature to record the Visual Basic commands to select a worksheet named HW17 and clear the contents of the entire sheet. Copy these commands into your homework 17 module. Next, use the Range("\$A\$" & i).formula properties to label values in column A and put the contents of the variables kwh, rate, and kwh*rate to cells B1, B2, and B3. Finally, use the record macro feature to record the Visual Basic commands to autofit the column width and format the values in cells B1, B2, B3. Copy these commands into your homework 17 module. Use a message box and GoTo statement as described above to either exit or continue.

Rate	Low KWH	High KWH
.0707	0	<= 400
.1414	400	<= 2000
.1808	2000	<= 4000
.2000	4000	

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

Always attempt the homework prior to the class it is due!

Homework is for class discussion only. It is not graded.

Never spend more than 45 minutes on the homework exercise!

If you can't understand it in 45 minutes, then bring your questions to class.

Homework 18: Visual Basic loop statements

This exercise provides practice working with Excel 5.0 Visual Basic. The Visual Basic program will be written on a module sheet, named homework18 and a worksheet, named HW18 will be used to display portions of the program output. The program simulates a coin toss. If the toss comes up heads, then the trial continues. If the toss comes up tails, then the trial ends. The macro calculates the maximum number of consecutive heads attained over a certain number of trials.

First, familiarize yourself with the syntax for the new commands used in this program by reading the following pages of the *Excel 5.0 Superbook* [For loops (pp. 696-698), while loops (pp. 695-696)]. Begin the module with the statement Option Explicit (p. 683). Define a sub procedure call homework18. Declare variables for the macro as well as the constants head=0 and tails=1. Use an InputBox command to prompt the user for number of trials. Next begin a for loop incremented from one to the number of trials. Set the counter for the number of consecutive heads to zero. Use the following if statement to simulate a coin flip within the body of the for loop.

```

If (Rnd() < 0.5) Then
    flip = head
    consecutive_heads = consecutive_heads + 1
Else
    flip = tail
End If
  
```

Use a while loop to continue flipping a coin while the flip is equal to heads. Count the number of consecutive heads. Outside the body of the while loop, determine whether the number of consecutive heads is greater than any prior number of consecutive heads attained. If so, update the maximum count. Continue to the next trial within the for loop.

Use the record macro feature to record the Visual Basic commands to select a worksheet named HW18 and clear the contents of the entire sheet. Copy these commands into your homework 18 module. On worksheet HW18, label cell A1 "number of trials". Put the value for the number of trials in cell B1. Label cell A2 "number of consecutive heads". Put the value for the number of consecutive heads in cell B2. Label cell A3 "observed probability". Put the value for observed probability in cell B3 [number of consecutive heads / number of trials]. Label cell A4 "actual probability". Put the value for actual probability in cell B4 [$0.5^{\text{max heads}}$] where actual probability = $0.5^{(\text{number of consecutive heads})}$. Finally, use the record macro feature to record the Visual Basic commands to autofit the column width and format the values in cells A1:B4. Copy these commands into your homework 18 module.

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 19: Visual Basic loop statements

This exercise provides practice working with Excel 5.0 Visual Basic. The Visual Basic program will be written on a module sheet, named homework19 and a worksheet, named HW19 will be used to display portions of the program output. The program simulates a coin toss. If the toss comes up heads, then the trial continues. If the toss comes up tails, then the trial ends. The macro calculates the maximum number of consecutive heads attained over a certain number of trials.

First, familiarize yourself with the syntax for the new commands used in this program by reading the following pages of the *Excel 5.0 Superbook* [For loops (pp. 696-698), while loops (pp. 695-696)]. Begin the module with the statement Option Explicit (p. 683). Define a sub procedure call homework19. Declare variables for the macro as well as the constants head=0 and tails=1. Use an InputBox command to prompt the user for number of trials. Next begin a for loop incremented from one to the number of trials. Set the counter for the number of consecutive heads to zero. Use the following if statement to simulate a coin flip within the body of the for loop.

```

If (Rnd() < 0.5) Then
    flip = head
    consecutive_heads = consecutive_heads + 1
Else
    flip = tail
End If

```

Use a while loop to continue flipping a coin while the flip is equal to heads. Count the number of consecutive heads. Outside the body of the while loop, determine whether the number of consecutive heads is greater than any prior number of consecutive heads attained. If so, update the maximum count. Continue to the next trial within the for loop.

Use the record macro feature to record the Visual Basic commands to select a worksheet named HW19 and clear the contents of the entire sheet. Copy these commands into your homework 19 module. On worksheet HW19, label cell A1 "number of trials". Put the value for the number of trials in cell B1. Label cell A2 "number of consecutive heads". Put the value for the number of consecutive heads in cell B2. Label cell A3 "observed probability". Put the value for observed probability in cell B3 [number of consecutive heads / number of trials]. Label cell A4 "actual probability". Put the value for actual probability in cell B4 [$0.5^{\text{max heads}}$] where actual probability = $0.5^{(\text{number of consecutive heads})}$. Finally, use the record macro feature to record the Visual Basic commands to autofit the column width and format the values in cells A1:B4. Copy these commands into your homework 19 module.

Homework Exercise

OPIM 101 Introduction to the Computer as an Analysis Tool

Homework Rules:

- Always attempt the homework prior to the class it is due!
- Homework is for class discussion only. It is not graded.
- Never spend more than 45 minutes on the homework exercise!
- If you can't understand it in 45 minutes, then bring your questions to class.

Homework 20: Visual Basic - arrays

This exercise provides practice working with Excel 5.0 Visual Basic. The Visual Basic program will be written on a module sheet, named homework20 and a worksheet, named HW20 will be used to display portions of the program output. The macro converts a dollar amount in cents into the maximum number of half-dollars, quarters, dimes, nickels and pennies that yield the equivalent dollar value. For example, the table below shows that \$1.43 is equal to two half- dollars, one quarter, one dime, one nickel and three pennies.

		Enter amount of change in cents:	143	
Value of coin =	50	Number of coins =	2	100
Value of coin =	25	Number of coins =	1	25
Value of coin =	10	Number of coins =	1	10
Value of coin =	5	Number of coins =	1	5
Value of coin =	1	Number of coins =	3	3
				143

Sub homework20()

Dim initial_amount As Integer

initialize

initial_amount = amount

final_formatting (initial_amount)

End Sub

Begin the module with the statement Option Explicit (p. 683). Define a sub procedure call homework20. Declare initial_amount as integer for the macro. Call a subroutine initialize, a function amount and a subroutine final_formatting. The subroutine, initialize, activates worksheet HW20 and clears the contents of the worksheet. The value initial_amount is passed to the subroutine, final_formatting. This subroutine autofits and formats the cells as shown in the table above, then the label 'Amount of change in cents:' is entered in cell C1 and the value in cell D1.

In the function amount, dimension an array named table and initialize the values to the value of each of the five coins. Use an InputBox command to prompt the user for the value of the change in cents. Use a for loop to compute the number of coins of each value that equals the initial amount in cents. Compute number of coins and amount as shown below:

quantity = Int(amount / table(index))

'number of coins calculation

amount = amount Mod table(index)

'modulus operator returns remainder of division

Spreadsheet Limitations

■ Spreadsheet limitations

- Each constant parameter can only assume one value at a time
- "What-if" analysis always results in a single point estimate

■ Monte Carlo simulation

- Allows a range of possible input values
- Statistical summary of outcome possibilities

Monte Carlo Simulation

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

Deterministic versus Probabilistic

■ Deterministic

- Makes use of average or "estimated" values for projected outcomes as though they were not subject to uncertainty
- Example:
How many units will we sell next year?
I'd estimate about 10,000.

■ Probabilistic

- Admits to uncertainty in projected outcomes
- Takes this quantified uncertainty into account in the subsequent analysis
- Example:
How many units will we sell next year?
I'd say somewhere between 8,500 and 12,000. Any value in that range has an equal chance of occurring.

Monte Carlo Simulation

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

Risk Analysis in Capital Budgeting

■ How large is the initial market?

- Expected value 250,000
-

■ How much can we charge for the product?

- Expected value \$510
-

■ How fast is the market growing?

- Expected value 3%
-

■ What will our market share be?

- Expected value 12%
-

Monte Carlo Simulation

OPIM 101



The Wharton School
of the University of Pennsylvania

Risk Analysis in Capital Budgeting

■ How large is the initial market?

- Expected value 250,000
- Range 100,000 - 340,000

■ How much can we charge for the product?

- Expected value \$510
- Range \$385 - \$575

■ How fast is the market growing?

- Expected value 3%
- Range .0.5% - 6.5%

■ What will our market share be?

- Expected value 12%
- Range 3% - 17%

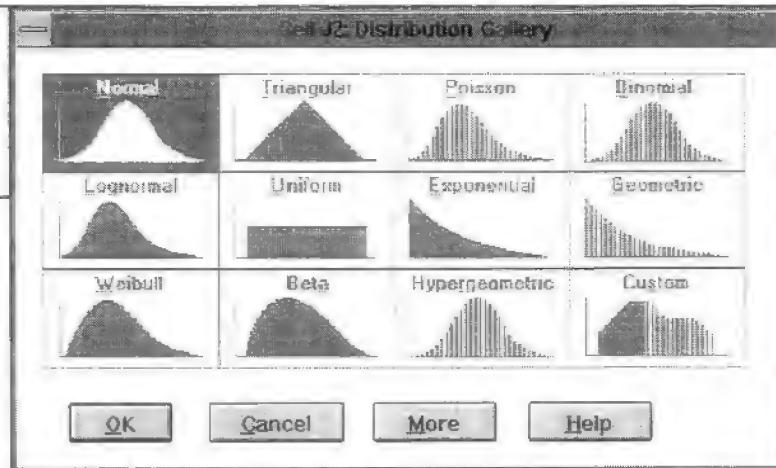
Monte Carlo Simulation

OPIM 101



The Wharton School
of the University of Pennsylvania

Common Probability Distributions



Monte Carlo Simulation

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

EXCEL Random number generation

RAND() Returns an evenly distributed random number greater than or equal to 0 and less than 1. A new random number is returned every time the worksheet is calculated.

Remarks To generate a random real number between a and b, use:

$\text{RAND}() * (b - a) + a$

If you want to use RAND to generate a random number but don't want the numbers to change every time the cell is calculated, you can enter =RAND() in the formula bar and press F9 (or COMMAND + = in Microsoft Excel for the Macintosh) to change the formula to a random number.

Example To generate a random number greater than or equal to 0 but less than 100: $\text{RAND}() * 100$

In Visual Basic use RND()

Monte Carlo Simulation

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Let us now consider
 the following example:

Let A be a matrix
 of order n and let
 B be a matrix
 of order m . Then
 the product AB is
 defined if and only
 if $n = m$.

Let us now consider
 the following example:
 Let A be a matrix
 of order n and let
 B be a matrix
 of order m . Then
 the product AB is
 defined if and only
 if $n = m$.

Behavioral Decision Making

People make decision every day

- Should I read the Primis book for OPIM 101 class?
- Should we launch the space shuttle Challenger?
- What kind of car should I buy?
- What apartment should I rent?

Three types of Decisions:

- Choices are well-defined alternatives to select
- Evaluations determine an amount (in \$) to bid
- Judgments are difficult to define an optimum or correct value (e.g. apartment selection)

Decision Making

OPIM 101



The Wharton School
of the University of Pennsylvania

Process Models of Choice

■ Noncompensatory rules

- Conjunctive rule - a strict cutoff is made for the salient dimensions (e.g., price < \$13,000 and MPG > 20; cars B, C, & D are not considered)
- Disjunctive rule - if one salient dimension meets the cutoff value, the choice is still considered even though another factor is above the cutoff. (e.g., price < \$13,000 or MPG > 20; only car D is not considered)

Car	Price	MPG	Safety
A	\$12,000	30	2
B	\$15,000	21	6
C	\$12,900	17	5
D	\$17,000	12	9

Decision Making

OPIM 101



The Wharton School
of the University of Pennsylvania

Process Models of Choice

■ Compensatory rules

- Trade-offs are made among the cutoff criteria (e.g., for each additional safety unit (above 5), I will spend \$1000 more for the car.

If safety < 5, then price \leq 14,000

If safety = 6, then price \leq 15,000

If safety = 7, then price \leq 16,000

If safety = 8, then price \leq 17,000

If safety = 9, then price \leq 18,000

C a r	P r i c e	M P G	S a f e t y
A	\$ 1 2 , 0 0 0	3 0	2
B	\$ 1 5 , 0 0 0	2 1	6
C	\$ 1 2 , 9 0 0	1 7	5
D	\$ 1 7 , 0 0 0	1 2	9

Decision Making

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Heuristics and Biases

- **Availability** over-estimate well-publicized events (probability of death from tornado versus lightening; more likely that word begins with "R" or has "r" as third letter of the word).
- **Representativeness** insensitive to base rate probabilities (farmers vs. NASA pilots); misconceptions of chance (HTHTTH is more random than HHHHHTH)
- **Anchoring and adjustment** insufficient adjustment from the anchor (initial ballpark estimate affects subsequent estimate)
- **Overconfidence bias** (OPIM 101 grades; driving skill - 69% of Sweden & 93% of USA drivers felt above average driver)
- **Selective perception** seek information consistent with their own view & fail to seek information to disconfirm their view

Decision Making

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Heuristics and Biases

- **Recency** give more attention to information heard last (ignore facts heard first)
- **Illusion of control** people pay more for a lottery ticket that they pick the number for than if randomly assigned by computer
- **Utility theory** risk averse in gain situations but risk seeking in loss situations
- **Time pressure** people generally make poorer quality decisions under stress conditions due to "satisficing"; also less risky under high time pressure
- **Group think** social pressures of the minority can unduly influence a majority

Decision Making

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Human Information Processing Limitations

- **Inability to manage decision process information**
 - STM constraints and unreliable recall of information
- **Difficulty in combining attributes or objectives**
 - STM limitations and slow numerical calculations
- **Inability to systematically search for "optimum"**
 - Time constraints, slow numerical calculations, and satisficing
- **Inaccuracies and biases in heuristic judgments**
 - Mental quantitative judgments are difficult
 - Unreliable recall of information
- **Generating and good problem representation**
 - Unable to create a mental representation

Decision Making

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Decision Models to Support Human Information Processing Limitations

- **Inability to manage decision process information**
 - DBMS can store, organize and retrieve data, but DM needs help keeping track of where they are in the DM process
- **Difficulty in combining attributes or objectives**
 - mathematical rules for combining attributes of decision outcomes can be used to aid the consistency of the DMs judgment policy
- **Inability to systematically search for "optimum"**
 - Statistical, DSS, AI tools help ease limited time constraints
- **Inaccuracies and biases in heuristic judgments**
 - DM awareness of their biases through feedback can help reduce the problem (e.g., bonuses for weather forecasters)
- **Generating and good problem representation**
 - Visual representation to aid problem solving

Decision Making

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Models of Decision Making

- **Rational view**
 - Theorems (Utility Theory) and optimal choice
 - Complete, perfect, and instantaneous information
- **Intuitive view**
 - Experience and human expertise
 - Verbal, experimental and historical information
- **Satisficing**
 - Rules of thumb (heuristics)
 - Sees the world as it is rather than as it should be
 - Uses a limited amount of information
 - Generates a few good alternatives, picking one that is good enough

Decision Making

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Decision Analysis

- Probability
- Expected value
- Decision trees
- Bayesian analysis and decision trees
- The value of information

Decision Analysis

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

Expected Value

- Suppose the following game is played:

Face Value	Payoff
1	\$2
2	\$5
3	\$5
4	\$10
5	\$10
6	-\$20

- Expected monetary value

$$\begin{aligned}
 \text{EMV} &= 1/6 \times \$2 + 1/6 \times \$5 + 1/6 \times \$5 + \\
 &\quad 1/6 \times \$10 + 1/6 \times \$10 - 1/6 \times \$20 \\
 &= \$12 / 6 = \$2
 \end{aligned}$$

Decision Analysis

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

Expected Value

- Should we advertise the new product if the $P(\text{hit}) = 0.2$ and $P(\text{flop}) = 0.8$?

Marketing	Hit	Flop
Advertise	\$15,000	- \$3,000
Do not advertise	\$3,000	- \$100

Payoff Matrix	State1	State2
Action1	Payoff11	Payoff12
Action2	Payoff21	Payoff22

- $EMV(\text{Advertise}) = .2 \times 15,000 - .8 \times 3,000 = \600
 $EMV(\text{No Advertise}) = .2 \times 3,000 - .8 \times 100 = \520

Decision Analysis

OPIM 101



The Wharton School
of the University of Pennsylvania

Expected Value

A contractor has been invited to bid on a construction job. The value of the contract depends on the length of time it takes to complete the project. If the project is finished on time, there is a profit of \$50,000. If the project is finished late, the contractor will lose \$10,000. Finishing late is solely dependent upon the weather. If the weather is good, the project will be finished on time. If the weather is bad, the project will be late. The contractor's subjective probability for good weather is 20%.

Should the contractor bid on the project?

Decision Analysis

OPIM 101



The Wharton School
of the University of Pennsylvania

Expected Value

Decision	Good Weather	Bad Weather
Bid	\$50,000	- \$10,000
Do not Bid	\$0	\$0
Probability	20%	80%

■ $EMV(\text{Bid}) = .2 \times 50,000 - .8 \times 10,000 = \$2,000$
 $EMV(\text{No not bid}) = .2 \times 0 - .8 \times 0 = \0

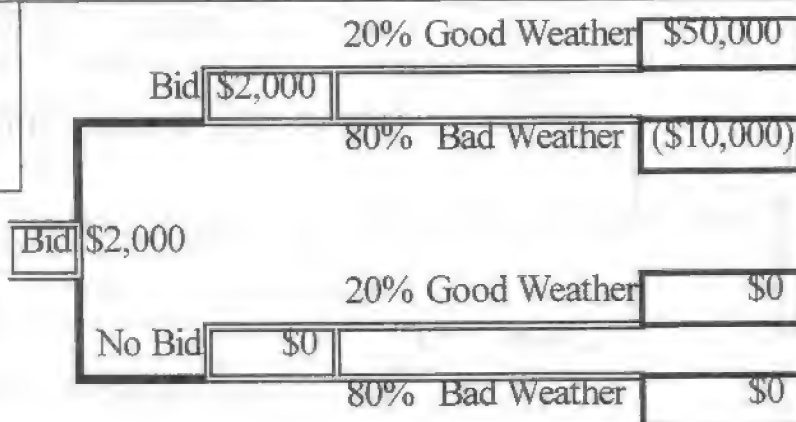
■ Therefore, bid on the project

Decision Analysis

OPIM 101

 **Wharton**
 The Wharton School
 of the University of Pennsylvania

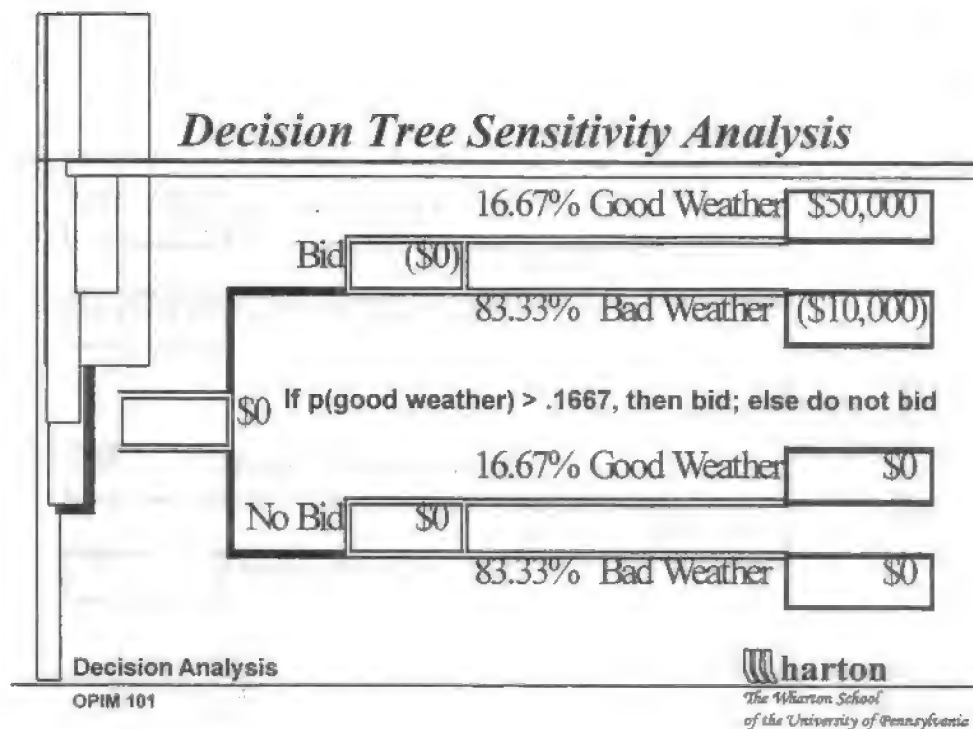
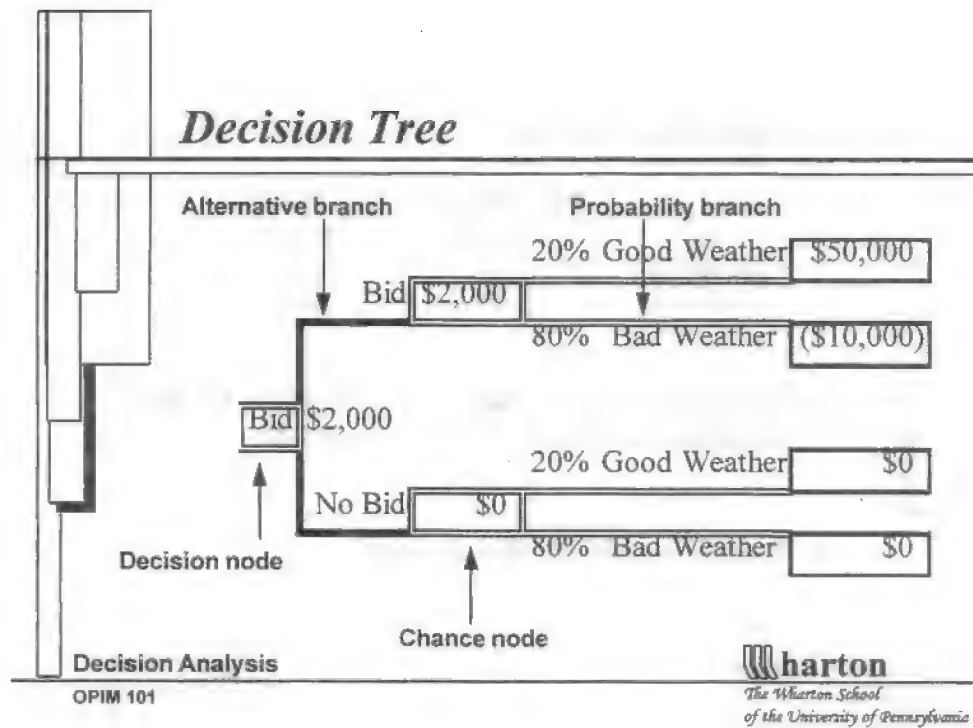
Decision Tree



Decision Analysis

OPIM 101

 **Wharton**
 The Wharton School
 of the University of Pennsylvania



Bayesian Analysis and Decision Trees

The contractor has an opportunity to buy a long-range forecast from an independent weather-forecasting company. The weather-forecasting company has a fairly good track record for these long-range forecasts. They successfully predicted good weather 70% of the time and 80% of the time it successfully predicted bad weather. The cost of the forecast is \$5,000.

	Weather is good	Weather is bad
Predict Good	.7	.2
Predict Bad	.3	.8

Decision Analysis

OPIM 101

 **Wharton**

The Wharton School
of the University of Pennsylvania

Bayesian Analysis and Decision Trees

(1)	(2)	(3)	(4)	(5)	(6)
State of Nature	Prior Probability	Conditional Probability Description	Conditional probability of sample	Cross product of prior X conditional probability	Probability posterior (5) divided by total of (5)
Good Weather	0.20	P(predict good good)	0.70	0.140	0.467
Bad Weather	0.80	P(predict good bad)	0.20	0.160	0.533
	1.00			0.300	1.000
Bad Weather	0.80	P(predict bad bad)	0.80	0.640	0.914
Good Weather	0.20	P(predict bad good)	0.30	0.060	0.086
	1.00			0.700	1.000

Decision Analysis

OPIM 101

 **Wharton**

The Wharton School
of the University of Pennsylvania

Bayesian Analysis and Decision Analysis

I_1 = prediction of good weather

I_2 = prediction of bad weather

S_1 = good weather

S_2 = bad weather

$$P(I_1 | S_1) = .7 \quad P(I_1 | S_2) = .2$$

$$P(I_2 | S_1) = .3 \quad P(I_2 | S_2) = .8$$

$$P(I_1) = P(I_1 | S_1) P(S_1) + P(I_1 | S_2) P(S_2) \\ = (.7)(.2) + (.2)(.8) = .30$$

$$P(I_2) = 1.0 - .30 = .7$$

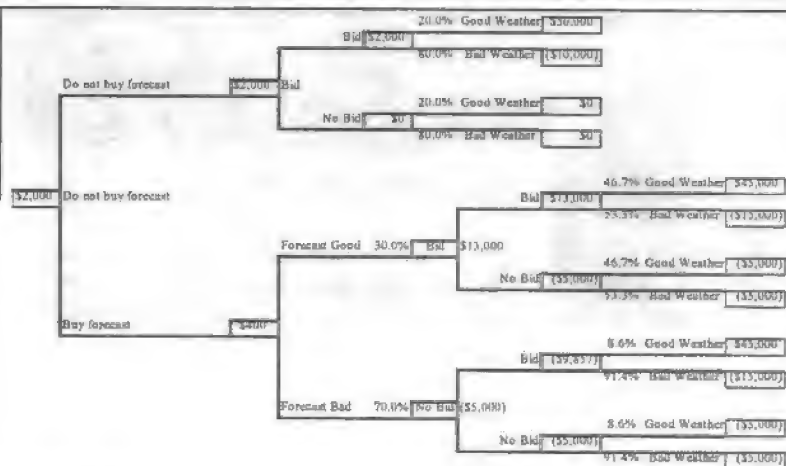
Decision Analysis

OPIM 101



The Wharton School
of the University of Pennsylvania

Bayesian Analysis and Decision Analysis



Decision Analysis

OPIM 101



The Wharton School
of the University of Pennsylvania

Bayesian Analysis and Decision Analysis

Contract Bid Decision Tree

GLL 24 Sep 94

Parameters	
Bid Good Weather	\$20,000
Bid Bad Weather	(\$10,000)
No Bid	\$0
P(Good Weather)	0.7
Weather Forecast Cost	\$5,000

Labels	
Good Weather	
Bad Weather	
Bid	
No Bid	

Conditional Probabilities	
Predict Good Weather	0.700
Predict Bad Weather	0.300

Joint Probabilities	
Predict Good Weather	0.490
Predict Bad Weather	0.210

Posterior Probabilities	
Predict Good Weather	0.457
Predict Bad Weather	0.543

Sample Probabilities	
Predict Good Weather	0.3
Predict Bad Weather	0.7

Decision Analysis

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Bayesian Analysis and Decision Analysis

Contract Bid Decision Tree

GLL 24 Sep 94

Results	
Do nothing	\$0
No Forecast EMV	\$2,000
Forecast EMV	\$400
EPPI	(\$10,000)
EVPI	\$8,000
EVSI	\$3,400

Decisions	
Do not buy forecast	
Bid	

Decision Analysis

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Value of Information

■ No forecast EMV

- Expected monetary value without sample information
- $[(.2)(\$50,000) + (.8)(\$10,000)] = \$10,000 - \$8,000 = \$2,000$

■ Forecast EMV

- Expected monetary value with sample information
- Bayesian decision analysis
- \$400
- Forecast EMV = \$0 if $p(\text{good weather}) = .186$

■ EPPI

- expected profit with perfect information
- always bid if good weather & never bid if bad weather
- $[(.2)(\$50,000) + (.8)(\$0)] = \$10,000$

Decision Analysis

OPIM 101



The Wharton School
of the University of Pennsylvania

Value of Information

■ EVPI

- Expected value of a perfect weather forecast
- EPPI minus no forecast EMV
- $[(.2)(\$50,000) + (.8)(\$0)] - \$2,000 = \$8,000$
- EVPI is the upper bound on the amount you would be willing to pay for a perfect weather forecast

■ EVSI

- Expected value of sample information
- EMV with sample information - EMV with no information
- If the cost of sample information has been subtracted from the payoff, then this amount must be added to the EMV with sample information
- $[\$400 + \$5,000] - \$2,000 = \$3,400$
- If price of weather forecast is negotiable, then this is most you are willing to pay for sample information

Decision Analysis

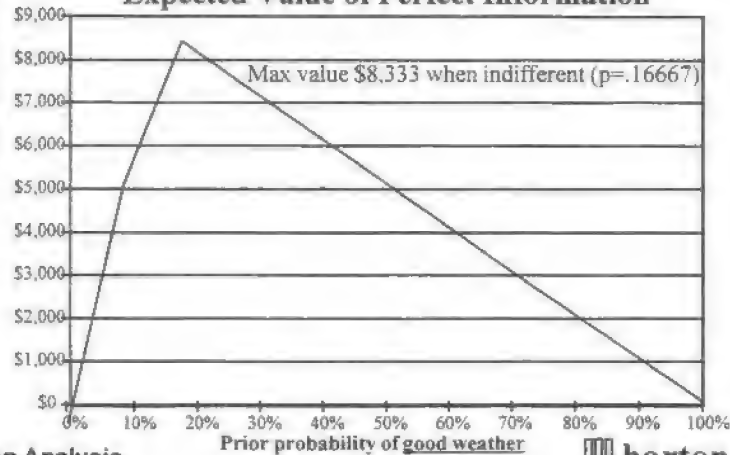
OPIM 101



The Wharton School
of the University of Pennsylvania

Value of Information

Expected Value of Perfect Information



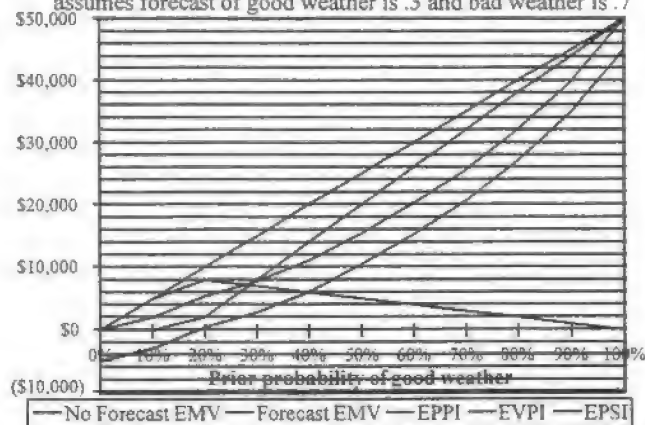
Decision Analysis

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Value of Information

Value of Information and the Probability of Good Weather assumes forecast of good weather is .3 and bad weather is .7



Decision Analysis

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania



Linear Programming

■ A mathematical technique for optimizing the use of resources given:

- an objective function expresses relationships among resource alternatives (e.g., minimize costs or maximize contribution)
- available resources are in limited supply
- alternative solutions could achieve the objective
- relationships must be expressed as linear equations or inequalities

■ Commonly used for

- mixture problems (Mrs. Fields Cookies, portfolio design, advertising in publications, manufacturing)
- routing problems (warehousing logistics, transportation)

Linear Programming

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Linear Programming Example

Champion Sports manufactures two types of custom men's underwear: boxers and briefs.

Briefs use 0.5 yards of material; boxers use 0.4 yards. 300 yards of material are available.

Each boxer uses 1 insignia logo and 600 insignia logos are in stock.

It requires 1 hour to manufacture one pair of boxers and 2 hours for one pair of briefs. 900 labor hours are available.

There is unlimited demand for boxers but total demand for briefs is 375 units per week.

The contribution per boxer is \$3.00 and the contribution per brief is \$4.50.

What mix maximizes contribution?

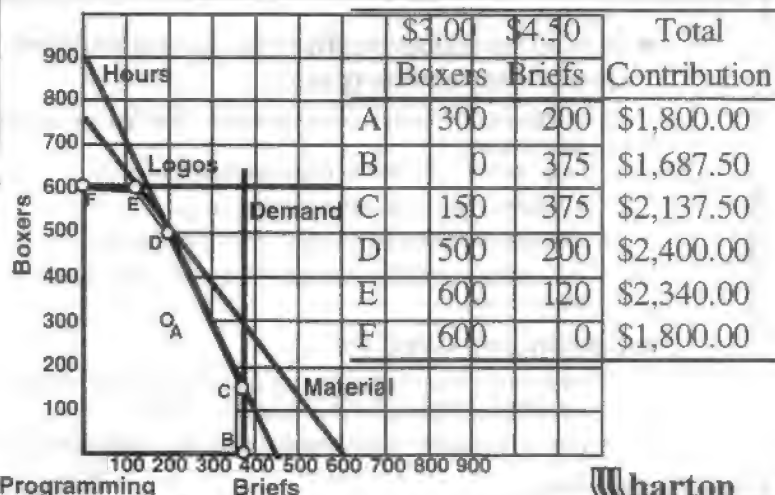
Linear Programming

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Test Corners for Maximum

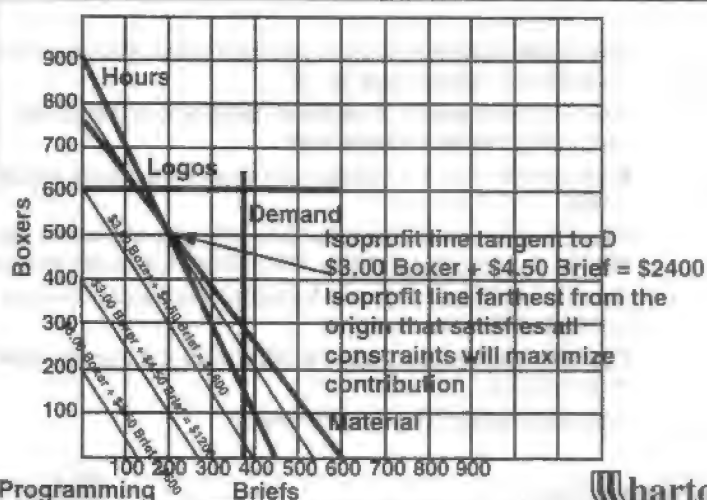


Linear Programming

OPIM 101

The Wharton School
of the University of Pennsylvania

Isoprofit Lines

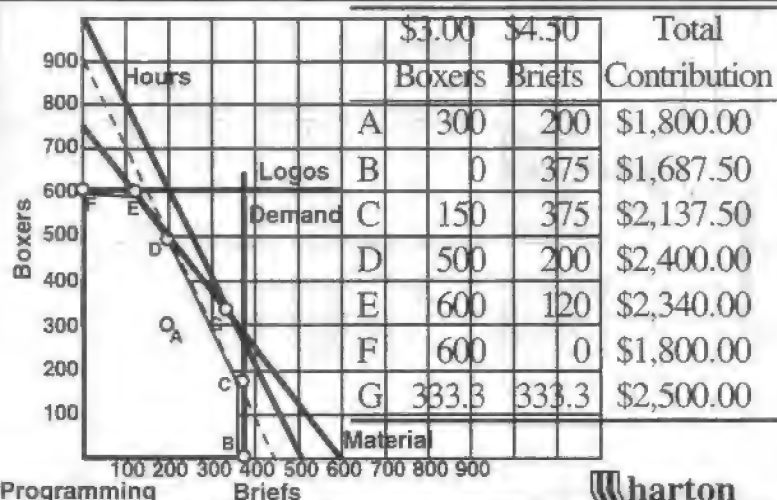


Linear Programming

OPIM 101

The Wharton School
of the University of Pennsylvania

*What if labor increased by 100 hours?
... the feasible region expands.*



Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Linear Programming Issues

• Extreme points

A corner of the feasible region formed by intersecting constraints. The solution is always at an extreme point.

• Infeasibility

There is no solution that satisfies each and every constraint. There is no feasible region.

• Unboundedness

The model has not been correctly formulated since the objective function can go to infinity without violating any constraint. Often missing a constraint.

• Redundancy

A constraint that does not affect the feasible region. Deleting redundant constraints enhances computational efficiency.

Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Excel Formulation

	A	B	C	D	E	F	G	H
1	Champion Sports							
2								
3		Boxers	Briefs	Usage	Maximum Resources			
4	Material	0.4	0.5	290	<=	300	yards	
5	Insignia Logos	1	0	600	<=	600	logos	
6	Labor	1	2	800	<=	900	hours	
7	Demand	0	1	100	<=	375	units	
8								
9	Production	600	100	Total				
10	Margin	\$3.00	\$4.50	\$2,250.00				
11								
12								

Linear Programming

OPIM 101

Wharton

Wharton

The Wharton School
of the University of Pennsylvania

Excel Solver

Excel Solver Parameters

Set Target Cell:

To: ☒ Max ☐ Min ☐ Value Of:

By Changing Variable Cells:

Subject to the Constraints:

Demand_Briefs <= \$F\$7

Hours_Labor <= \$F\$6

Logos <= \$F\$5

Yards_Material <= \$F\$4

Solve

Close

Guess

Options...

Add...

Change...

Delete

Reset All

Help

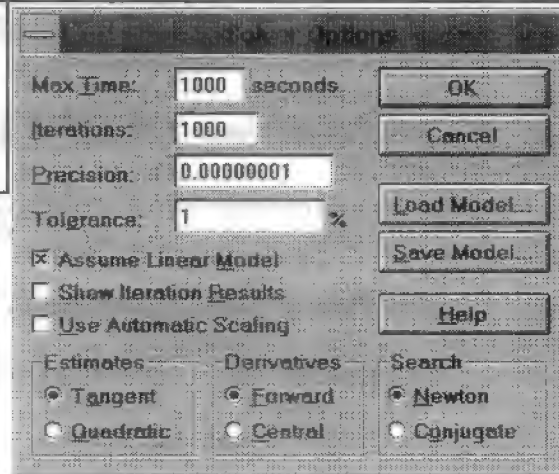
Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Excel Solver

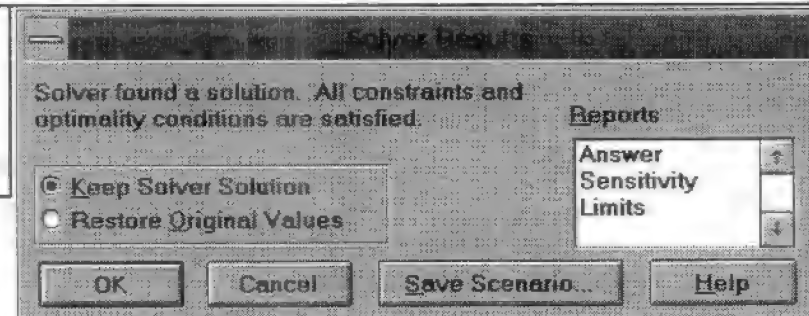


Linear Programming

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Excel Solver



- Press the Ctrl key and use the mouse to select both the Answer and Sensitivity Reports

Linear Programming

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Excel Solver

■ The Solver could not find a feasible solution

- This is usually an unboundedness or infeasibility problem.

■ The maximum iteration or time limit was reached: continue anyway?

- The warning prevents infinite computer time for an unsolvable model. A Solver Options button allows you to increase these limits.

■ If the Solver solution does not look correct,

(e.g., 2 decision variables and 1 binding constraint)

- Try increasing the tolerance to 1% under the Solver Options Menu.

Linear Programming

OPIM 101



The Wharton School
of the University of Pennsylvania

Excel Solver

■ Assume linear model option

- Greatly speeds computation if model is linear
- Generates error message if model is not linear
- Must specify this option for right-hand-side and objective ranging
- If not selected, Solver generates a Sensitivity Report using Reduced Gradient instead of Reduced Cost and Lagrange Multiplier instead of Shadow Price

Linear Programming

OPIM 101



The Wharton School
of the University of Pennsylvania

Excel Solver Answer Report

Microsoft Excel 5.0 Answer Report
Worksheet: [94BOXERS.XLS]Boxers Base Case
Report Created: 8/24/94 9:22

Target Cell (Max) Optimum objective function value

Cell	Name	Original Value	Final Value
\$D\$10	Margin Total	\$2,400.00	\$2,400.00

Adjustable Cells Optimum decision variable values

Cell	Name	Original Value	Final Value
\$B\$9	Boxer_Production	500	500
\$C\$9	Brief_Production	200	200

Constraints Status of the constraints

Cell	Name	Cell Value	Formula	Status	Slack
\$D\$4	Yards_Material	300	\$D\$4<=\$F\$4	Binding	0
\$D\$5	Logos	500	\$D\$5<=\$F\$5	Not Binding	100
\$D\$6	Hours_Labor	900	\$D\$6<=\$F\$6	Binding	0
\$D\$7	Demand_Briefs	200	\$D\$7<=\$F\$7	Not Binding	175
\$B\$9	Boxer_Production	500	\$B\$9>=0	Not Binding	500
\$C\$9	Brief_Production	200	\$C\$9>=0	Not Binding	200

Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Excel Solver Answer Report

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
\$D\$4	Yards_Material	300	\$D\$4<=\$F\$4	Binding	0
\$D\$5	Logos	500	\$D\$5<=\$F\$5	Not Binding	100
\$D\$6	Hours_Labor	900	\$D\$6<=\$F\$6	Binding	0
\$D\$7	Demand_Briefs	200	\$D\$7<=\$F\$7	Not Binding	175
\$B\$9	Boxer_Production	500	\$B\$9>=0	Not Binding	500
\$C\$9	Brief_Production	200	\$C\$9>=0	Not Binding	200

■ Status of constraints

- binding constraints never have slack or surplus
- not binding constraints always have slack or surplus
- not satisfied (incorrect specification of the constraints)
- number of binding constraints \geq number of decision variables

■ Slack measures unused available resources

Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Interpreting LP Solutions

- **Shadow price** - value of 1 additional unit of that resource
change in optimal objective function value
 unit increase in right-hand-side coefficient
- **Right-hand-side ranging**
 upper and lower boundary range over which shadow prices are valid. Multiple RHS changes are possible.
- **Reduced cost** - value of not using 1 unit of that resource
change in optimal objective function value
 unit increase of that non-basic variable (unused)

Linear Programming

OPIM 101



The Wharton School
of the University of Pennsylvania

Interpreting LP Solutions

- **Objective ranging**
 For each objective function coefficient, there is an upper and lower boundary range of values over which the optimal solution to the problem does not change.

 As the value of an objective coefficient changes, the optimal objective function value, the shadow prices, and the reduce costs will change, however the values of the optimal basic (used in solution) variables do not change.

 Objective ranging provides a sensitivity analysis of how the solution changes as we move past the bounds of the original optimal solution. However, to obtain the exact solution, the model must be resolved.

Linear Programming

OPIM 101



The Wharton School
of the University of Pennsylvania

Excel Solver Sensitivity Report

Microsoft Excel 5.0 Sensitivity Report
Worksheet: [94BOXERS.XLS]Boxers Base Case
Report Created: 8/24/94 9:23

Changing Cells ← Reduced cost and objective ranging

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
\$B\$9	Boxer_Production	500	\$0.00	3	0.6	0.75
\$C\$9	Brief_Production	200	\$0.00	4.5	1.5	0.75

Constraints ← Shadow prices and right-hand-side ranging

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$D\$4	Yards_Material	300	\$5.00	300	15	52.5
\$D\$5	Logos	500	\$0.00	600	1E+30	100
\$D\$6	Hours_Labor	900	\$1.00	900	131.25	60
\$D\$7	Demand_Briefs	200	\$0.00	375	1E+30	175

Linear Programming

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Excel Solver Sensitivity Report

■ Shadow prices and right-hand-side ranging

Constraints Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$D\$4	Yards_Material	300	\$5.00	300	15	52.5
\$D\$5	Logos	500	\$0.00	600	1E+30	100
\$D\$6	Hours_Labor	900	\$1.00	900	131.25	60
\$D\$7	Demand_Briefs	200	\$0.00	375	1E+30	175

❶ How much would you pay for one additional insignia logo?

Nothing since logos are not constraining the solution!

Linear Programming

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Excel Solver Reports

- How much would you pay for one additional insignia logo?
- A new market survey shows that demand will double from 375 to 750. How many additional briefs should be manufactured to meet this new demand?
- A stain is found on 15 yards of material reducing material from 300 to 285 yards. How does this affect total contribution?
- Labor is willing to negotiate 100 additional hours of production work. How much should management be willing to pay?
- If labor offers 100 additional hours of production work at \$1.50 per hour should management be willing to accept the offer?
- Product designers are considering offering a new line of padded briefs that require 1 yard of material and 2 hours of labor. Contribution would be \$6.00 per padded brief. Should management begin this new line?

Linear Programming

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Excel Solver Reports

- If material decreases from 300 to 290 yards and labor increases from 900 to 1000 hours, what is the change in weekly contribution?
- A management consultant offers to improve efficiency in the production of boxers. The improvements will increase the contribution by \$0.50 to \$3.50. What is the new mix? What is the increase in weekly contribution?
- The contribution for briefs decreases by \$0.75 to \$3.75. What is the new mix? What is the decrease in weekly contribution?

Linear Programming

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Simplex Method

- **Graphic method is limited to two decision variables**
- **More complex models use Simplex method**
 - algebraic representation of extreme points (constraint corners)
 - algorithm proceeds from extreme point to adjacent extreme point (each move is called an iteration)
 - if unbounded, simplex method discovers this during execution
 - For a maximization problem, simplex algorithm will generally increase for each iteration or decrease for each iteration of a minimization problem
- **Extremely large problems use Karmarker LP**
 - by Narendra Karmarker of AT&T Bell Laboratories in 1984
 - Desert Storm logistics - 500,000 variables & 70,000 constraints

Linear Programming

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

The Dual in Linear Programming

- **Duality examines the value of resources corresponding to the constraints in the original or primal model (e.g. What is the value of 1 additional hour of labor?) .**
- **There is one dual variable for each constraint.**
- **The dual variable is zero for all non-binding constraints.**
- **In Champion Sports, the dual LP minimizes use of hours labor, material, logos, and demand subject to the constraint that the value of the resources used is greater than or equal to its contribution margin**

Linear Programming

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Excel Formulation of the Dual

	A	B	C	D	E	F	G	H
1		Champion Sports						
2								
3		Material	Insignia Logos	Labor	Demand	MINIMIZE		
4	Cost / unit	\$5.00000	\$0.00000	\$1.00000	\$0.00000			Contrib
5	Max Resource	300.00	600.00	900.00	375.00	\$2,400.00000		Margin
6	Boxers	0.40	1.00	1.00	0.00	\$3.00	>=	\$3.00
7	Briefs	0.50	0.00	2.00	1.00	\$4.50	>=	\$4.50
8		yards	units	hours	units			
9								

Note formula in F6 =SUMPRODUCT(\$B\$4:\$E\$4,B6:E6)

Linear Programming

OPIM 101



Wharton

The Wharton School
of the University of Pennsylvania

Excel Formulation of the Dual

■ Answer report showing binding constraints

- There are 4 binding constraints
- Material and Labor that were binding in the primal are not binding in the dual

Cell	Name	Cell Value	Formula	Status	Slack
\$F\$6	Boxer Contribution	\$3.00	\$F\$6>=\$H\$6	Binding	\$0.00
\$F\$7	Brief Contribution	\$4.50	\$F\$7>=\$H\$7	Binding	\$0.00
\$B\$4	Material	\$5.00000	\$B\$4>=0	Not Binding	\$5.00000
\$C\$4	Insignia Logos	\$0.00000	\$C\$4>=0	Binding	\$0.00000
\$D\$4	Labor	\$1.00000	\$D\$4>=0	Not Binding	\$1.00000
\$E\$4	Demand	\$0.00000	\$E\$4>=0	Binding	\$0.00000

Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Excel Formulation of the Dual

Changing Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
\$B\$4	Material	\$5.00000	0.00	300	15.00	52.50
\$C\$4	Insignia Logos	\$0.00000	100.00	600	1E+30	100
\$D\$4	Labor	\$1.00000	0.00	900	131.25	60.00
\$E\$4	Demand	\$0.00000	175.00	375	1E+30	175

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$F\$6	Boxer contribution	\$3.00	500.00	\$3.00	\$0.60	\$0.75
\$F\$7	Brief contribution	\$4.50	200.00	\$4.50	\$1.50	\$0.75

Sensitivity report showing shadow prices and reduced costs for the dual LP

Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Interpretations of Duality

- Resource "rents" must be at least as much from producing.

$$(\$3.00 \times \text{Boxers}) + (\$4.50 \times \text{Briefs})$$

MUST BE LESS THAN OR EQUAL TO

$$300 \times \text{Material} + 600 \times \text{Logos} + 900 \times \text{Hours} + 375 \times \text{Demand}$$

- Optimal value is always the same in both the primal and its dual (\$2,400 in this example)
- Simplex method solves the primal and dual simultaneously

- Values of shadow prices in one optimal solution (primal or dual) are always equal to the value of the structural variables in the other optimal solution (primal or dual).
- Thus, all information from the dual is available from the primal

Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Application: Red Brand Canners

Contract price for the 3 million pound tomato crop was \$.06 per pound. 80% of the crop was grade B; 20% of the crop was grade A.

There was demand for 14.4 million pounds whole tomatoes, 1 million pounds juice tomatoes, and 2 million pounds paste tomatoes.

A quality scale is used to rate tomatoes on a 0 - 10 point scale (Ten is best). Grade A averages 9 points per pound; grade B averages 5 points per pound. The minimum grade quality for whole tomatoes is 8 points, and the minimum grade quality for juice is 6 points.

The margin for whole, juice, and paste tomatoes was \$1.48, \$1.32, and \$1.85 per case with 18, 20, and 25 pounds of tomatoes per case, respectively.

What mix will maximize contributions at Red Brand Canners?

Linear Programming

OPIM 101



Wharton

*The Wharton School
of the University of Pennsylvania*

Application: Red Brand Canners

- Define the 6 decision variables:
- Whole grade A
- Whole grade B
- Juice grade A
- Juice grade B
- Paste grade A
- Paste grade B
- What are the non-negativity constraints?



Linear Programming

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Application: Red Brand Canners

What are the constraints for supply of tomatoes?

Let WA = whole grade A
WB = whole grade B
JA = juice grade A
JB = juice grade B
PA = paste grade A
PB = paste grade B

Contract price for the 3 million pound tomato crop was \$.06 per pound. 80% of the crop was grade B; 20% of the crop was grade A.



Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Application: Red Brand Canners

What are the constraints for demand of tomatoes?

Let WA = whole grade A
WB = whole grade B
JA = juice grade A
JB = juice grade B
PA = paste grade A
PB = paste grade B

There was demand for 14.4 million pounds whole tomatoes, 1 million pounds juice tomatoes, and 2 million pounds paste tomatoes.



Linear Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Application: Red Brand Canners

What are the constraints for quality of tomatoes?

A quality scale is used to rate tomatoes on a 0 - 10 point scale (Ten is best). Grade A averages 9 points per pound; grade B averages 5 points per pound. The minimum grade quality for whole tomatoes is 8 points, and the minimum grade quality for juice is 6 points.



Linear Programming

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Application: Red Brand Canners

What is the objective function?

Let WA = whole grade A
 WB = whole grade B
 JA = juice grade A
 JB = juice grade B
 PA = paste grade A
 PB = paste grade B

The margin for whole, juice, and paste tomatoes was \$1.48, \$1.32, and \$1.85 per case with 18, 20, and 25 pounds of tomatoes per case, respectively.



Linear Programming

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Neural Networks

- Analogy between neural nets and the nervous system
- History of neural networks
- How neural nets work
- Example problem
- Common questions about neural networks
- Application examples
- Selected references
- Summary



Neural Networks

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Analogy between neural nets and the nervous system

- **Neural nets based on nodes and connections**
Analogous to a nerve cell - 10^{12} neurons and 10^{14} synaptic connections in the human brain
- **Nodes have input signals**
Dendrites carry an impulse to the neuron
- **Nodes have one output signal**
Axons carry signal out of neuron and synapses are local regions where signals are transmitted from the axon of one neuron to dendrites of another.
- **Input signal weights are summed at each node**
Nerve impulses are binary; they are "go" or "no go". Neurons sum up the incoming signal and fire if a threshold value is reached.



Neural Networks

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

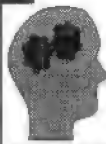
How Neural Nets Work

■ Implementation

- Hardware - electronic circuits mimic neurons
- Software - linkages of nodes, inputs, and outputs can be programmed

■ Uses a trial and error method of learning

- Finds patterns associating inputs and outputs using a large set of training data where both inputs and outputs are known (e.g. use the intermarket relationship among the Standard & Poor's 500 index, 30-year Treasury bonds, and the commodity research bureau index to predict direction of the S&P 500 index trend 5 weeks into the future)
- Initially begins with random weights and corrects mistakes by modifying the weight that it has given each input item.



Neural Networks

OPIM 101



The Wharton School
of the University of Pennsylvania

How Neural Nets Work

■ Feedback network

- A given node's output can be transmitted back to itself or to other previous nodes as another input

■ Feedforward network

- All outputs only go forward

■ Parallel distributed processing versus serial symbolic processing

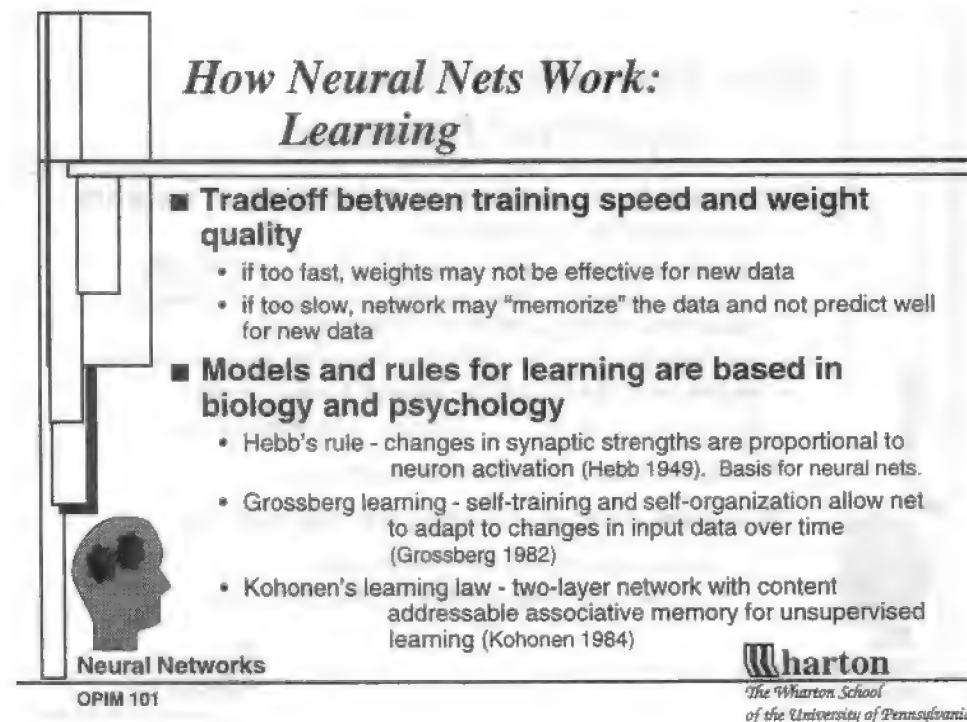
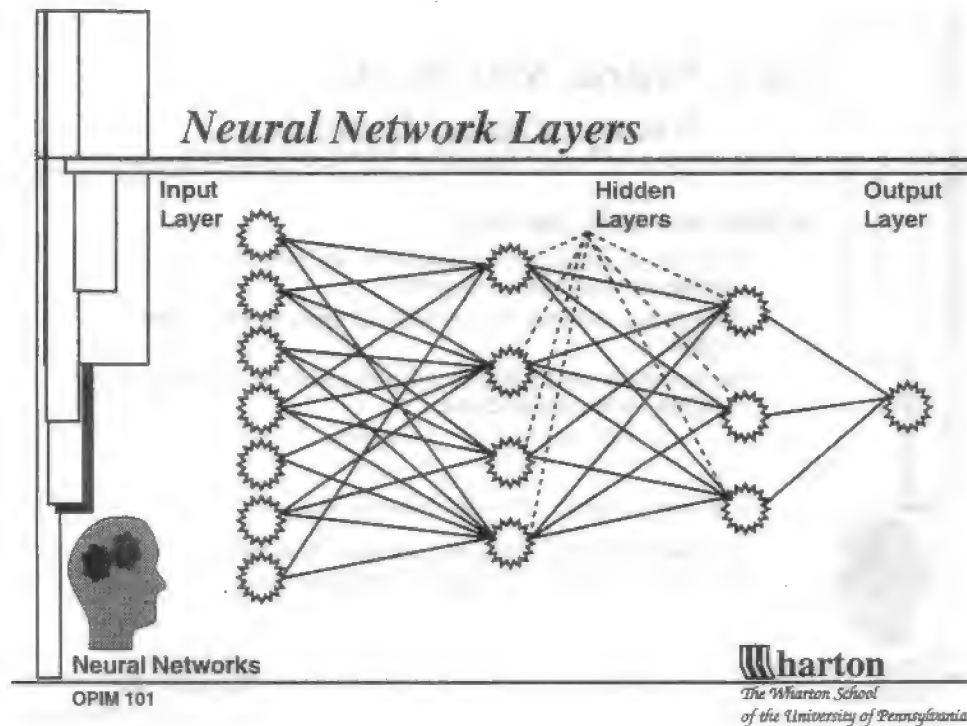


Neural Networks

OPIM 101



The Wharton School
of the University of Pennsylvania



How Neural Nets Work: Unsupervised Learning

■ Nets are self-learning

- BAM (bi-directional associative memory) used for OCR, speller checker, voice recognition
- Weight adjustments are not from comparison with known values
- Based on the input pattern, only weights for the winning node or a few nodes are modified

$$\Delta W_{ij} = A_i A_j \quad \text{where:}$$

A_i is the activation of the i th node in one layer

A_j is the activation of the j th node in another layer

W_{ij} is the connection strength between two nodes



Neural Networks

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

How Neural Nets Work: Supervised Learning

■ Gradually train weights to meet desired outputs

- inputs presented to the network
- weights adjusted to achieve desired output for training data
- corrections based on difference between actual and desired output which is computed for each training cycle
- if average error is within tolerance- stop, else continue training
- weights are locked in and the network is ready to use

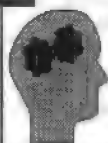
$$\Delta W_{ij} = \alpha A_i (C_j - B_j) \quad \text{where } \alpha \text{ is the learning rate,}$$

A_i is the activation of the i th node in one layer

B_j is actual activation of the j th node in recalled pattern,

C_j is desired activation of the j th node, and

W_{ij} is the connection strength between two nodes



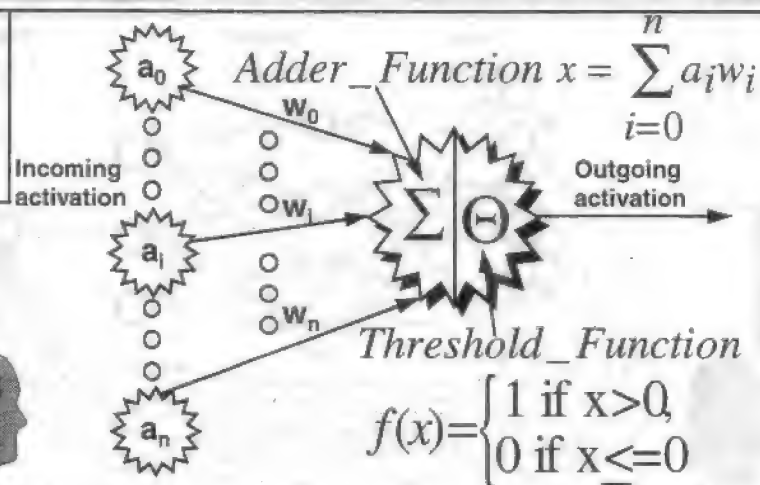
Neural Networks

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Mathematical Model of a Node



Neural Networks

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

How Neural Nets Work Back Propagation

- ① Input is presented to net and output is produced
- ② Compute differences between actual and desired outputs
- ③ Adjust output layer weights using discrepancies between desired outputs and actual outputs
- ④ Then adjust hidden layer weights (if there is a hidden layer)
- ⑤ Then adjust input layer weights
- ⑥ Repeat steps 1 - 5 until desired accuracy level is achieved

■ Advantage:

- ability to learn any arbitrarily complex nonlinear mapping

■ Disadvantages:

- extremely long - potentially infinite - learning times
- Speed up using parallel hardware

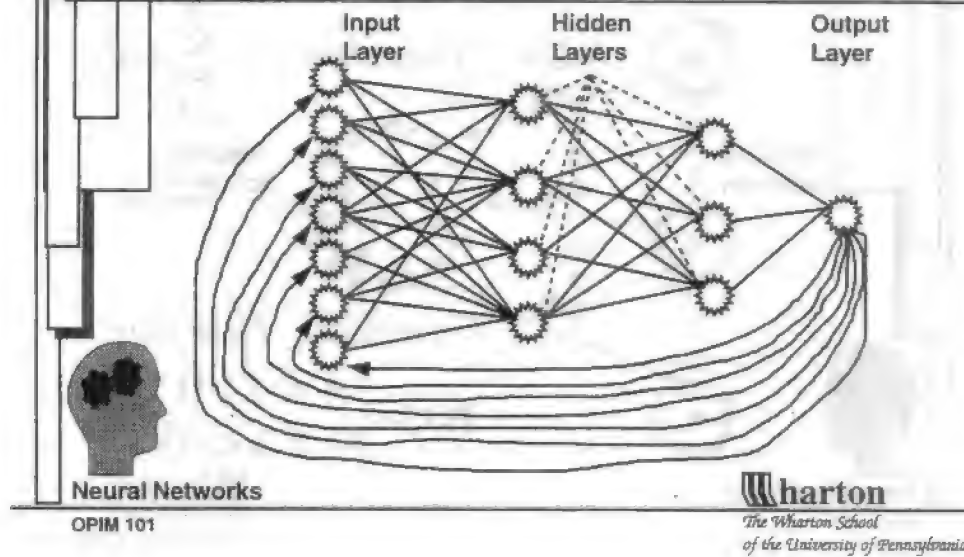
Neural Networks

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Neural Network Layers: Back Propagation



Common Questions About Neural Networks

■ What is a hidden layer?

- Group of nodes between the input and output layer
- Hidden layers increase the ability of the network to "memorize" the data

■ How many hidden layers should I use?

- As problem complexity increases, number of hidden layers should also increase
- Start with none. Add hidden layers one at a time if training or testing results do not achieve target accuracy levels

■ What is a hidden node?

- A node in a hidden layer is called a hidden node
- A hidden node contains much of the knowledge in the network and act as filters to remove noise moving through the network

Neural Networks
OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Common Questions About Neural Networks

■ How many nodes and hidden nodes should I use?

- Ideally, you will have between 2 and 3 training cases per connection (synapse). Fewer training cases per connection will cause problems generalizing the test data.
- For example, suppose you have 60 inputs, 240 training cases and one hidden layer with 5 hidden nodes. A fully connected network would have $60 \times 5 + 5$ connections (305). This is less than one training case per connection (240/305).
- Correct by decreasing inputs to between 15 and 23 (determine which 15 or 23 by trial and error)
 - $15 \times 5 + 5$ connections (80) for a 3:1 ratio (240/80)
 - $23 \times 5 + 5$ connections (120) for a 2:1 ratio (240/120)
- Or correct by reducing the number of hidden nodes to 2. A fully connected network would have $60 \times 2 + 2$ connections (122). This is almost 2 training cases per connection (240/122).



Neural Networks

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Common Questions About Neural Networks

■ How do I know if network modifications are needed?

- Low accuracy of training or test data indicates that a new hidden layer or more hidden nodes are needed
 - if number of hidden nodes exceeds number of inputs and outputs, then add another hidden layer
 - decrease the total hidden nodes by 50% in each successive hidden layer [if 10 nodes in first layer, then use 5 in the second layer and 2 in the third layer]
- If Braincel performs well on the Training and Test ranges, but poorly on new records, then it is treating each record as a special case and has "memorized" the data
 - use fewer hidden nodes or remove the hidden layer
- Could also need more training cases per connection



Neural Networks

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Application Examples: Finance and Banking

- **Firm failure prediction** (Koster, Sondak, & Bourbia 1991; Wilson & Sharda 1993)
- **Bank failure prediction** (Cinar & Lash 1992; Tam & King 1992)
- **Bond rating** (Utans & Moody 1991)
- **Mortgage credit approval** (Reilly et al. 1990)
- **Credit card fraud prevention** at Chase Manhattan Bank, American Express, and Mellon Bank examine unusual credit-charge patterns over a history of usage and compute a fraud potential rating. [For example, the Fraud Detection System by Nestor Corp. and a system by HNC Inc. (Rochester 1990)].
- **Takeover target prediction** (Sen & Gibbs 1992)



Neural Networks

OPIM 101



The Wharton School
of the University of Pennsylvania

Application Examples: Finance and Banking

- **Country risk rating for early warning of financial risk** (Roy and Cosset 1990)
- **Stock price prediction** (Fishman, Barr, & Loick 1991; Yoon & Stein 1991)
- **Commodity, futures, and currency trading** at Merrill Lynch, Salomon Brothers, Shearson Lehman Brothers, & the World Bank. Citibank claims 25% returns in currency trading using GA trained neural nets (Business Week March 2, 1992)
- **Asset allocation** (Steiger & Sharda 1991)
- **Corporate merger prediction** (Sen, Oliver, & Sen 1992)



Neural Networks

OPIM 101



The Wharton School
of the University of Pennsylvania

Application Examples: Manufacturing

- Quality control
- Predict tool breakage in milling operations
- Force and / or wear analysis
- Mechanical equipment fault diagnosis
- Process management and control - maintain efficiency of electric arc furnaces in steel-making; uniformity in pulp & paper process management



Neural Networks

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Application Examples: Marketing

- Customer mailing list management (Hall 1992)
- Spiegel Inc. mail order catalog targets saved \$1 million from reduced costs and increased sales (Business Week March 2, 1992)
- Airline seating allocation and passenger demand for Nationair Canada and US Air (IEEE Expert Dec 1992)
- Customer purchasing behavior and merchandising-mix strategies
- Hotel room pricing - yield management (Relihan, W. 1989)



Neural Networks

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

PROPOSAL FOR A NEW BOOK

THE NEW BOOK

THE NEW BOOK

The new book is a collection of essays on the history of the United States. It is written by a group of leading historians and is intended for a general audience. The book covers the period from the early colonial years to the present. It is a comprehensive and accessible work that provides a clear and concise overview of the history of the United States.

PROPOSAL FOR A NEW BOOK

THE NEW BOOK

The new book is a collection of essays on the history of the United States. It is written by a group of leading historians and is intended for a general audience. The book covers the period from the early colonial years to the present. It is a comprehensive and accessible work that provides a clear and concise overview of the history of the United States.

The new book is a collection of essays on the history of the United States. It is written by a group of leading historians and is intended for a general audience. The book covers the period from the early colonial years to the present. It is a comprehensive and accessible work that provides a clear and concise overview of the history of the United States.

The new book is a collection of essays on the history of the United States. It is written by a group of leading historians and is intended for a general audience. The book covers the period from the early colonial years to the present. It is a comprehensive and accessible work that provides a clear and concise overview of the history of the United States.

Genetic Algorithms

- Problem size
- What are genetic algorithms?
- Genetic algorithm components
- Example problem
- Common questions about genetic algorithms
- Example applications
- Important references for genetic algorithms
- Ten summary points



Genetic Algorithms

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

How large is the decision space?

- If we were to look at every alternative, what would we have to do? Of course, it depends.....
- Think: enzymes
 - Catalyze all reactions in the cell
 - Biological enzymes are composed of amino acids
 - There are 20 naturally-occurring amino acids
 - Easily, enzymes are 1000 amino acids long
 - $20^{1000} = (2^{1000})(10^{1000}) \approx 10^{1300}$

- A reference number, a benchmark:

$10^{80} \approx$ number of atomic particles in universe



Genetic Algorithms

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Heuristic Search

■ Build rule-based expert systems

- Performance so far not super impressive (somewhat impressive)
- Doesn't show what's needed. Only shows that there exist such rules, not how they are found or how cognition could work. (rule-governed vs rule-described)
- Expensive and very time-consuming in general

■ Build programs that acquire rules automatically

- Genetic algorithms
- Performance so far is very impressive (e.g., suspect ID)
- Still time-consuming, but can hope for a general architecture

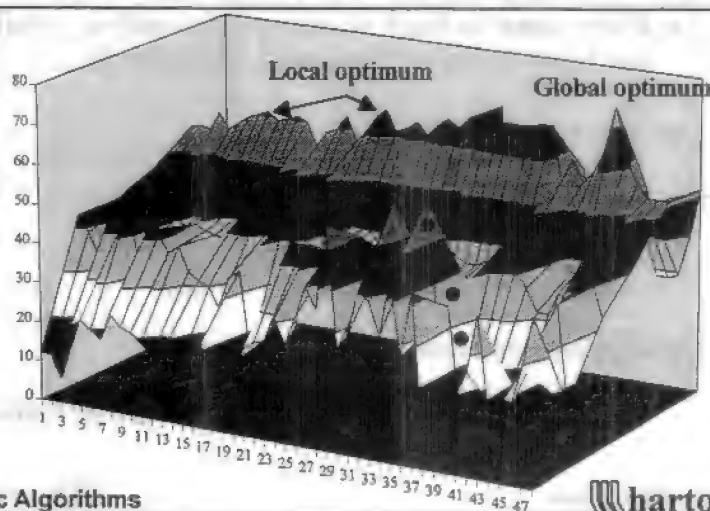
Genetic Algorithms

OPIM 101

harton

The Wharton School
of the University of Pennsylvania

Genetic algorithms vs hill climbing



Genetic Algorithms

OPIM 101

harton

The Wharton School
of the University of Pennsylvania

Genetic Algorithm Components

■ Selection

- determines how many and which individuals breed
- premature convergence sacrifices solution quality for speed

■ Crossover

- select a random crossover point
- successfully exchange substructures
- 00000 x 11111 at point 2 yields 00111 and 11000

■ Mutation

- random changes in the genetic material (bit pattern)
- for problems with billions of local optima, mutations help find the global optimum solution

■ Evaluator function

- rank fitness of each individual in the population
- simple function (maximum) or complex function

Genetic Algorithms

OPIM 101

 **harton**
The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Example Problem

Annual sales for Avoiding Extinction by JWI Publishers is 20,000 copies. Books are sold for \$30.

JWI Publishers have a variable cost of \$6 per book associated with producing the book.

JWI Publishers have two fixed cost components.

Overhead, royalties and other costs total \$350,000.

Setup cost per printing is \$6,000. Thus, 4 quarterly printing would cost $4 \times \$6,000 = \$24,000$.

Genetic Algorithms

OPIM 101

 **harton**
The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Example Problem

Annual sales for Avoiding Extinction by JWI Publishers is 20,000 copies. Books are sold for \$30.

JWI Publishers have a variable cost of \$6 per book associated with producing the book.

JWI Publishers have two fixed cost components.

Overhead, royalties and other costs total \$350,000.

Setup cost per printing is \$6,000. Thus, 4 quarterly printing would cost $4 \times \$6,000 = \$24,000$.

EOQ model yields

$$N_{unit} = \sqrt{\frac{2PU}{C}} = \sqrt{\frac{2 * \$6,000 * 20,000}{\$6}}$$

$$N_{unit} = 6,325 \text{ books / order}$$

Genetic Algorithms

OPIM 101

harton

The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Example Problem

Quantity	6,326
Annual Book Sales	20,000
Number of Setups	3.16
Setup Cost	\$6,000
Selling Price	\$30
Variable Book Costs	\$6

Total Revenues	\$600,000
Variable Costs	\$18,978
Fixed Costs	\$18,969
Other Costs	\$350,000
Profit	\$212,052.67

Genetic Algorithms

OPIM 101

harton

The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Example Problem

① Choose the problem representation

- 14 digits binary string allows an order size from 1 to 16,384 (2^{14})

② Initialize the population

- randomly generate 100 - 200 individual strings of length 14

③ Calculate fitness for each individual

- convert string to decimal and determine profit with that order size
- 00100010011010 = 2,202

Total Revenue	\$600,000
Variable costs	$6 * 2,202 / 2$
Setup costs	$20,000 / 2,202 * \$6,000$
Fixed costs	\$350,000
Profit	\$188,898

Genetic Algorithms

OPIM 101

harton
The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Example Problem

④ Perform selection

- long run survival of the fittest
- short run merely nudges population towards better performers
- replace the worst strings (bottom 5%) with copies of the best strings (top 5%), thus it would take a minimum of 20 generations before all strings are replaced - slow convergence.

⑤ Perform crossover

- randomly select two parents from the new population
- randomly determine whether to crossover ($p \approx .6$)
- if crossover, randomly select a crossover point (1-13)
- example:
00100010011010 (2,202) x 11011001000111 (13,895)
at 3 yields
11000010011010 (3,655) x 00111001000111 (12,442)

Genetic Algorithms

OPIM 101

harton
The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Example Problem

⑥ Perform mutation

- bit by bit, string by string, randomly determine whether to mutate each bit using a very low probability ($p=.007$). If mutation rate is too high, it will prevent convergence.
- if mutation should occur, change 0 to 1 or vice versa.

⑦ Check convergence

- bias is one measure of agreement among the population
- bias assumes values between 50 and 100 percent
- bit bias
 - if 100 strings have 0 in position 1 and 100 have a 1, then the bit bias is 50%
 - a 75:25 split or a 25:75 split has a 75% bias
 - a 90:10 split or a 10:90 split has a 90% bias
- string bias is the average bias for each bit over all strings
- a population with a average bias of 95% has converged

Genetic Algorithms

OPIIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

Common Questions about Genetic Algorithms

■ Can a GA converge to a poor solution?

YES! Poor problem representation, premature convergence, a poor fitness evaluation algorithm, or luck of the random numbers could generate a poor solution

■ How do you know whether the GA solution is optimal or near optimal?

If you knew how to find the optimal solution, you would not need to use a GA. There is no guarantee that a GA will find an optimal solution. GAs find a good solution that is "better" than others.

■ Are neural networks better than GAs?

Neural networks require less structural knowledge. However, the type and number of node connections and hidden layers make it difficult to interpret relationships in a neural network.

GAs require a starting framework to setup the problem representation and calculate fitness

Genetic Algorithms

OPIIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

Important References for Genetic Algorithms

- Holland, J.H. 1975. Adaptation in natural and artificial systems. Ann Arbor, MI: The University of Michigan Press.
 - classic technical book with lots of theorems and proofs
- Goldberg, D.E. 1989. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.
 - graduate textbook for a machine learning course - code in Pascal
- Davis, L. (ed) 1991. Handbook of genetic algorithms. New York: Van Nostrand Reinhold.
 - tutorial and case applications with code in C or Lisp

Genetic Algorithms

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Important References for Genetic Algorithms

- Koza, J. 1992. Genetic programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press.
 - application of programs as bits rather than 1s and 0s
- Bauer, R.J. Jr. 1994. Genetic algorithms and investment strategies. New York: Wiley.
 - examples of GAs used for trading bonds and stocks
- Karjalainen, R. and Allen F. 1994. Using genetic algorithms to find technical trading rules. Wharton working paper

Genetic Algorithms

OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Genetic Algorithms: Summary

- ① Field is not new. Holland's work began in 1970s.
- ② Most of the work has been done in computer science & engineering - not business applications
- ③ Translate problem into a string representation - often binary numbers (11000)
- ④ Difficult to perform translation for some problems
- ⑤ Little knowledge at startup - randomly generated population of individuals



Genetic Algorithms

OPIM 101

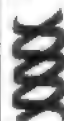


The Wharton School

of the University of Pennsylvania

Genetic Algorithms: Summary

- ⑥ Must be able to calculate fitness of each individual in the population
- ⑦ Potential solutions with greater fitness have greater priority in subsequent generations
- ⑧ Crossover is similar to mating and mutation transforms a stable population to maintain diversity of the search process
- ⑨ Steps 6-8 repeat. Eventually the population converges and the fittest solution survives
- ⑩ GAs are not an optimization technique but often find good solutions for large complex problems



Genetic Algorithms

OPIM 101



The Wharton School

of the University of Pennsylvania

Functions of a Database System

- Store and organize data efficiently
- Create and maintain the database
- Provide information to users
- Provide tools to meet changing information requirements
- Protect the database against inconsistency and errors
- Safeguard the database against unauthorized access

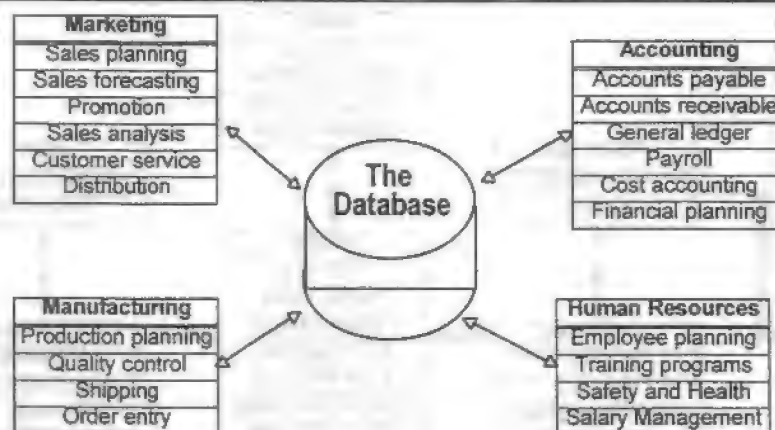
Database Systems

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Database Systems



Database Systems

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Data Representation

- Bit binary digit 1 or 0
- Byte 1 alpha-numeric character
- Field or column customer no., name, address, order number
- Record or row all customer fields in 1 row
- File or data set all customers
- Data base customers, sales staff, inventory
- File processing
- Database management

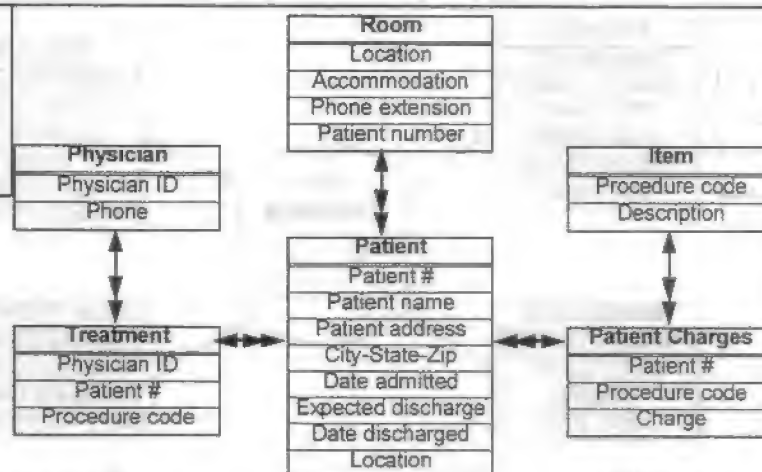
Database Systems

OPIM 101



The Wharton School
of the University of Pennsylvania

Database Systems



Database Systems

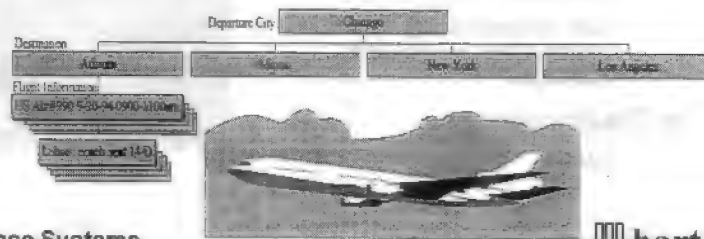
OPIM 101



The Wharton School
of the University of Pennsylvania

Hierarchical

- Relationships defined at database creation
- Advantages
 - two times faster than relational
 - simple to understand and model
- Disadvantages
 - can not pose ad hoc queries



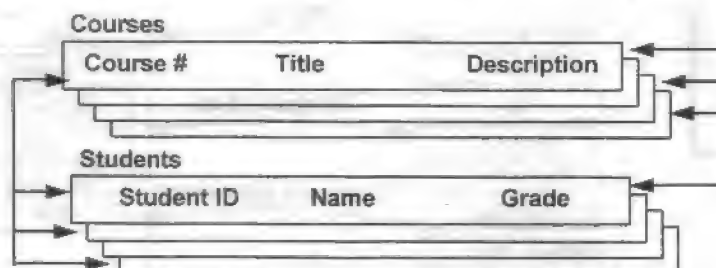
Database Systems

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Network

- Subordinate records linked to multiple parents
- 1:1, 1:M, and M:N relationships
- Complex and difficult to apply



Database Systems

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Relational

■ Properties

- each row is a record
- each column is a field
- each relation is a 2D table

■ Advantage

- best for ad hoc queries
- can represent any hierarchical or network database

■ Disadvantage

- Slow

Flight	From	To	Depart	Arrive
US 126	Chicago	New York	730	830
US 872	Chicago	Miami	800	1100
US 990	Chicago	Atlanta	800	1000
AA 1260	Chicago	New York	1030	1130
AA 435	Chicago	Miami	1600	1900
NW 1542	Chicago	Atlanta	1600	1830

Name	Flight	Seat
Lohse	US 990	C 14-D
Smith	US 990	1st 2-A
Jones	US 990	C 15-A
Moore	US 990	C 26-C
Hurst	US 990	C 31-B
Yu	US 990	C 11-A

Database Systems

OPIM 101



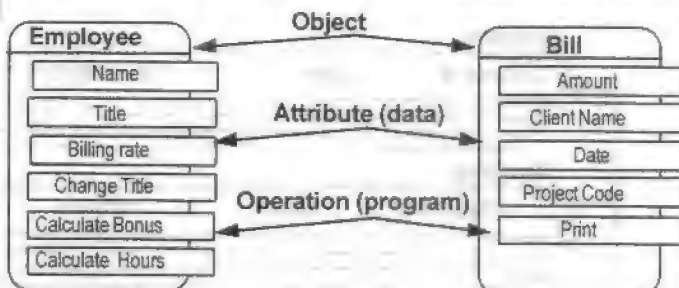
The Wharton School
of the University of Pennsylvania

Object-Oriented

■ Objects are specific entities

■ Classes are program defined groups of objects

■ Inheritance of all or some attributes or operations (e.g. full-time vs part-time employee)



Database Systems

OPIM 101



The Wharton School
of the University of Pennsylvania

Database Systems

■ Data integrity - accuracy, correctness, & validity of data

- Safeguard data against invalid alteration or destruction
- Concurrent user access with record locking prevents one user from accessing a record while another is updating the same record.
- Avoid redundant data

■ Data independence

- physical - modify physical scheme without need to rewrite application programs
- logical - modify conceptual scheme without causing application programs to be rewritten

Database Systems

OPIM 101

 **harton**
The Wharton School
of the University of Pennsylvania

Relational Database Terminology

- **Relation:** a table of tuples and attributes
- **Tuple:** corresponds to a row of a table
- **Attribute:** corresponds to a column or field
- **Cardinality:** the number of tuples in a table
- **Degree:** the number of attributes in a table
- **Primary Key:** unique identifier for every record in the table - no null values!
- **Candidate Key:** any key that can serve as a primary key

Database Systems

OPIM 101

 **harton**
The Wharton School
of the University of Pennsylvania

Relational Database Terminology

- **Composite Primary Key:** combination of primary keys to make a unique identifier
- **Foreign Key:** attribute in one table that is a primary key in another table
- **Nonkey attribute:** any attribute that is not part of the primary key

Database Systems

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database Unnormalized

Student			Course		Instructor		Grade
Student #	Name	Major	Course #	Title	Name	Location	
38214	Bright	IS	IS 350	Database	Codd	B104	A
			IS 465	Sys Anal	Kemp	B213	C
69173	Smith	MGT	MGT 101	Intro	Regis	B309	A
			HIS 200	Am Civil	Jones	L3091	B
			MGT 104	HRM	Staff	B320	B
42356	WU	FIN	FIN 203	Corp Fin	Siegel	S203	A
			FIN 204	Banking	Inman	S223	A
			FIN 207	Sec Anal	Herring	S224	B

- **Course number is a repeating group!**

Database Systems

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Normalization

- Basically normalization eliminates repeating groups to avoid redundancy in all tables
- Principal normal forms:
 - 1NF - all primary keys are defined, all nonkey attributes depend on the primary key, and no repeating groups
 - 2NF - no partial dependencies (only possible for tables with a composite primary key)
 - 3NF - no transitive dependencies
 - BC (Boyce/Codd) - equivalent to 3NF unless multiple candidate keys, composite keys, and candidate keys have at least one attribute in common.

Database Systems

OPIM 101

 **harton**

The Wharton School
of the University of Pennsylvania

The Registrar Database - 1NF

Student #	Course #	Course	Instructor		Grade
		Title	Name	Location	
38214	IS 350	Database	Codd	B104	A
	IS 465	Sys Anal	Kemp	B213	C
69173	MGT 101	Intro	Regis	B309	A
	HIS 200	Am Civil	Jones	L3091	B
	MGT 104	HRM	Staff	B320	B
42356	FIN 203	Corp Fin	Siegel	S203	A
	FIN 204	Banking	Inman	S223	A
	FIN 207	Sec Anal	Herring	S224	B

3NF Student		Major
Student #	Name	
38214	Bright	IS
69173	Smith	MGT
42356	WU	FIN

- No repeating groups! 1NF
Student # Course #

- Student # is 3NF

Database Systems

OPIM 101

 **harton**

The Wharton School
of the University of Pennsylvania

The Registrar Database - 1NF

■ Insertion Anomaly

- Insert OPIM 101, Introduction to Computers with a new course number and new course title
- One student must register for OPIM 101 to be able to insert!
- Same problem with adding an instructor!

■ Update Anomaly

- Change title from Sys Anal to Sys Anal & Des
- User must search through all tuples and update course each time it occurs!

■ Deletion Anomaly

- If one student is enrolled in an independent study and drops the course, we also lose information about the title and the instructor!

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database - 2NF

2NF		Course	Instructor	Instructor
Course #	Title	Name	Location	
IS 350	Database	Codd	B104	
IS 465	Sys Anal	Kemp	B213	
MGT 101	Intro	Regis	B309	
HIS 200	Am Civil	Jones	L3091	
MGT 104	HRM	Staff	B320	
FIN 203	Corp Fin	Siegel	S203	
FIN 204	Banking	Inman	S223	
FIN 207	Sec Anal	Herring	S224	

3NF		
Student #	Course #	Grade
38214	IS 350	A
	IS 465	C
69173	MGT 101	A
	HIS 200	B
	MGT 104	B
42356	FIN 203	A
	FIN 204	A
	FIN 207	B

3NF		
Student #	Name	Major
38214	Bright	IS
69173	Smith	MGT
42356	WU	FIN

■ No partial dependencies

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database - 2NF

■ Insertion Anomaly

- Insert new instructor
- Can not insert instructor without assigning her a course!

■ Update Anomaly

- Each instructor can teach multiple courses
- User must search through all tuples and update instructor data each time it occurs!

■ Deletion Anomaly

- If one course is deleted, we also lose information about the instructor!

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database - 3NF

3NF			3NF			Instructor	
Course #	Title	Instructor Name	Student #	Course #	Grade	Name	Location
IS 350	Database	Codd	38214	IS 350	A	Codd	B104
IS 465	Sys Anal	Kemp		IS 465	C	Kemp	B213
MGT 101	Intro	Regis	69173	MGT 101	A	Regis	B309
HIS 200	Am Civil	Jones		HIS 200	B	Jones	L3091
MGT 104	HRM	Staff		MGT 104	B	Staff	B320
FIN 203	Corp Fin	Siegel	42356	FIN 203	A	Siegel	S203
FIN 204	Banking	Inman		FIN 204	A	Inman	S223
FIN 207	Sec Anal	Herring		FIN 207	B	Herring	S224

■ No transitive dependencies

3NF		
Student #	Name	Major
38214	Bright	IS
69173	Smith	MGT
42356	WU	FIN

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database - 3NF

- 2NF anomalies concerning insertion, deletion, and updating are removed.
 - There may still be anomalies when a relation has multiple primary keys
 - Boyce-Codd normal form
- No information is lost during the normalization process
- Redundant information is reduced

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database - Boyce-Codd normal form

3NF		
Student	Major	Advisor
123	Physics	Bohr
123	Music	Mozart
456	Biology	Darwin
789	Physics	Bohr
999	Physics	Einstein

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Registrar Database - Boyce-Codd normal form

BCNF		BCNF	
Student	Advisor	Advisor	Major
123	Bohr	Bohr	Physics
123	Mozart	Mozart	Music
456	Darwin	Darwin	Biology
789	Bohr	Einstein	Physics
999	Einstein		

- Equivalent to 3NF if the primary key is not a composite

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Parts-Supplier Database Suppliers

S	S#	SNAME	STATUS	CITY
S1	Smith	20	London	
S2	Jones	10	Paris	
S3	Blake	30	Paris	
S4	Smith	20	London	
S5	Adams	30	Athens	

- Why two suppliers with SNAME of Smith?
- What fields are candidate keys for this relation?
- What does STATUS mean? How would you find out?
- Ordering on the rows?

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Parts-Supplier Database - Parts

P	P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London	
P4	Screw	Red	14	London	
P6	Cog	Red	19	London	
P2	Bolt	Green	17	Paris	
P5	Cam	Blue	12	Paris	
P3	Screw	Blue	17	Rome	

- Does CITY in P mean the same as CITY in S?
 - Suppliers are located in s.city
 - Parts are stored in p.city
 - If P3 was not stored in Rome, does that mean there is not a supplier in Rome?
- 17 what? Pounds? Ounces? Tons? Kilograms?
- How would you find all parts supplied in Paris?

Database Systems

OPIM 101



The Wharton School
of the University of Pennsylvania

The Parts-Supplier Database - Orders

SP	S#	P#	QTY
	S1	P2	200
	S1	P3	400
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P4	300

- Double key, S#-P#
- Why more than one table in the Parts-Supplier database?
- How do we pose queries that rely on data in more than one table?

Database Systems

OPIM 101



The Wharton School
of the University of Pennsylvania

Structured Query Language - SQL

■ General form of SELECT statement in Access:

```
SELECT DISTINCTROW <attributes to be displayed>
FROM <tables>
WHERE <conditions are met>
ORDERED BY <attribute> [DESC]
```

- Use of square brackets, e.g., P.[P#] for the P# field of table P
- DISTINCTROW avoids duplicate records
- ALL means duplicate records are not removed

Database Systems

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Structured Query Language Example 1

```
SELECT DISTINCTROW P.COLOR, P.CITY
FROM P
WHERE P.CITY <> "Paris"
AND P.WEIGHT > 10 ;
```

- What does this say?
- Note: ORDERED BY is optional and here absent
- Returns a table listing the color and city of the part for all parts in the relation P where the city is not equal to "Paris" and the part weight is greater than 10

Database Systems

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Strategic Applications

- Quality control at Whirlpool Corporation
- Commercial marine paint sales
- American Airlines Saber reservation system
- Rosenbluth Travel
- American Hospital Supplies



Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Information Retrieval Problem

- The IR problem is very hard
- Why? Many reasons, including:
 - Documents are not (very) structured
 - Database searches vs document base searches
 - Language is not (very) cooperative
 - DNA: microbiology or DEC Network Architecture?
 - Free rider: game theory or urban transportation systems?
 - Corporate memory or organizational memory?
- Physical access vs logical access
 - Physical: relatively easy
 - Logical: terribly difficult

Database Systems

OPIM 101



 **harton**

*The Wharton School
of the University of Pennsylvania*

The Information Retrieval Problem: Basic IR Technology

■ Your basic IR technology

- Full text or keyword retrieval, with
- Boolean combinations and
- Location indicators

■ Full text--has everything

- Or does it?

■ Keyword indexing

- Requires work

■ Boolean combination of words

- Usual Boolean operators: AND, OR, NOT
- This is a logically complete set



Database Systems

OPIM 101

harton

*The Wharton School
of the University of Pennsylvania*

The Information Retrieval Problem: Probability of Retrieving a Relevant Document

$P(\text{word}_1) = .6$ probability searcher uses word_1 in a query

$P(\text{word}_2) = .5$ probability searcher uses word_2 in a query

$P(\text{Doc_word}_1) = .7$ probability word_1 is in relevant document

$P(\text{Doc_word}_2) = .6$ probability word_2 is in relevant document

The probability of searcher using word_1 in a query and word_1 being in a relevant document is $P(\text{word}_1) \times P(\text{Doc_word}_1) = .6 \times .7 = .42$

The probability of searcher using word_1 in a query and word_2 being in a relevant document is $P(\text{word}_1) \times P(\text{Doc_word}_2) = .6 \times .6 = .36$

The probability of searcher using word_1 and word_2 in a query and both word_1 and word_2 being in a relevant document is $P(\text{word}_1) \times P(\text{word}_2) \times P(\text{Doc_word}_1) \times P(\text{Doc_word}_2) = .6 \times .5 \times .7 \times .6 = .126$

Database Systems

OPIM 101

harton

*The Wharton School
of the University of Pennsylvania*

The Information Retrieval Problem: Basic IR Technology

	Relevant	Not Relevant	
Retrieved	X	U	Total Number Retrieved = n_1
Not Retrieved	V	Y	
	Total Number Relevant = n_2		

Recall measures how well all relevant documents are retrieved (x / n_2)

Precision measures how well only relevant documents are retrieved (x / n_1)

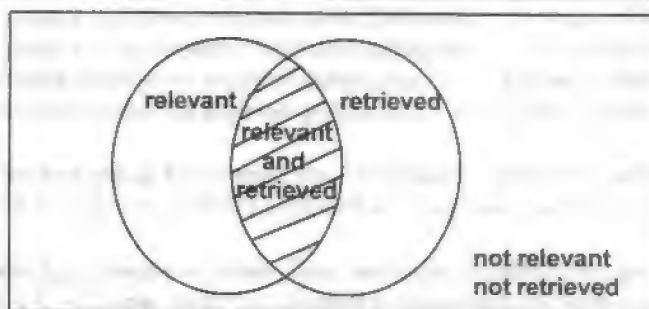
Database Systems

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

The Information Retrieval Problem: Basic IR Technology



- When and where and how does the recall vs precision distinction matter?
- How well does full text retrieval work?

Database Systems

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

The Information Retrieval Problem: Summary of Blair and Maron Study

- Searcher perception that their search was exhaustive (recall > 75%) actual recall 20%
- No significant difference between searching ability of lawyer or paralegal
- Searchers were only able to anticipate a small number of words and phrases that could be used to retrieve relevant documents and would not be in irrelevant documents
- Extraordinary and unpredictable variability in the words and phrases used to discuss the same topics (e.g., the accident in the litigation referred to as situation, difficulty, event, what happened last week, and we all know why we are here)

Database Systems

OPIM 101

 **harton**

*The Wharton School
of the University of Pennsylvania*

The Information Retrieval Problem: Why is IR such a difficult problem?

- Zipfian word distributions (plot of index words by rank gives a hyperbolic shape with long tails)
- Scale is the problem
- Concept: futility point(s)
- Demise of the library model
- Collection partitioning
- IR as communication
- Importance of context

Database Systems

OPIM 101



 **harton**

*The Wharton School
of the University of Pennsylvania*

The first of these is the fact that the
 world is not a uniform whole. It is
 divided into many different parts, each
 with its own characteristics and its own
 history. This is true of the physical world
 as well as of the human world. The
 second is the fact that the world is
 constantly changing. It is not a static
 entity, but a dynamic one. The third
 is the fact that the world is not
 a single entity, but a collection of
 many different entities. Each of these
 entities has its own life and its own
 death. The world is a complex of
 many different parts, each with its
 own life and its own death. The world
 is a collection of many different
 entities, each with its own life and
 its own death. The world is a
 collection of many different entities,

each with its own life and its own
 death. The world is a collection of
 many different entities, each with
 its own life and its own death. The
 world is a collection of many different
 entities, each with its own life and
 its own death. The world is a
 collection of many different entities,
 each with its own life and its own
 death. The world is a collection of
 many different entities, each with
 its own life and its own death. The
 world is a collection of many different
 entities, each with its own life and
 its own death. The world is a
 collection of many different entities,

What is a macro?

- Program to execute repetitive
- Perform special calculations
- Create a custom user interface

Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Programming with Visual Basic

```

BOOK1.XLS
Function Commissions (SharesSold, PricePerShare)
    TotalSalePrice = SharesSold * PricePerShare
    If TotalSalePrice <= 15000 Then
        Commissions = 25 + .03 * SharesSold
    Else
        Commissions = 25 + .03 * (.9 * SharesSold)
    End If
End Function
  
```

Sheet1

Module1

Programming

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Programming with Visual Basic

BOOK1.XLS				
C5		=Commissions(C3,C4)		
	A	B	C	D
1	Stock Data			
2		Original Price	\$24	
3		Number of Shares	100	
4		Sale Price	\$55	
5		Commission	\$28	

Programming

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Programming with Visual Basic

■ Macro Advantages

1. Faster operations
2. Reduce human involvement in repetitive tasks
3. Fewer errors
4. Fewer keystrokes per operation
5. Enhanced features and interface options
6. Wider calculation options

■ Macro Disadvantages

1. Development time
2. Maintenance and support work over time
3. Uses macrosheet and worksheet
4. Forget to update information using macro
5. Macrosheet must be open to use the macro

Programming

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Alternatives to Macros

- 1. Templates
- 2. Command shortcut keys
- 3. Using style sheets in addition to templates
- 4. Use worksheet parameters

Programming
OPIM 101

 **Wharton**
*The Wharton School
of the University of Pennsylvania*

Two Types Of Macros

- **Command macros**
 - execute menu commands
 - programmed using the recorder
 - user initiates the action - Ctrl-b
- **Function macros**
 - programmed =average()
 - e.g. formula paste functions
 - functions for special calculations
 - worksheet formula initiates macro

Programming
OPIM 101

 **Wharton**
*The Wharton School
of the University of Pennsylvania*

Local versus Global Macros

■ Local macros

- module **MUST** be open

■ Global macros

- stored on a hidden macro sheet called `global.xlm`
- automatically loaded with Excel

Programming

OPIM 101



*The Wharton School
of the University of Pennsylvania*

Macro Examples

■ Command macro for formatting borders

■ Function macro for converting temperature

■ Adding a new menu bar

■ Interface for marketing forecasts

■ Database for student class registration

Programming

OPIM 101



*The Wharton School
of the University of Pennsylvania*

Distinct Parts of Function Macros

- **Workbook name (if in a different workbook)**
- **Macrosheet name**
- **Separate macrosheet & function with !**
- **Name of function**
- **List of one or more function arguments**
 - example: [workbook.xls]demo!FtoC2(B4)

Programming

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Debugging Visual Basic Code

- **Syntax errors**
 - Misspelled or missing keywords and punctuation marks
 - Most are caught as you type (Intteger vs. Integer)
- **Compile errors**
 - Syntax errors found only when you run the program
 - If ... Then statement without an End If
 - Generates an error message
- **Runtime errors**
 - Errors found only when you run the program
 - Division by zero or using a property with wrong object
- **Logic errors**
 - No error message - output is not what you expected
 - Use trace procedures - very difficult to debug

Programming

OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Debugging Visual Basic Code

■ Setting a breakpoint in Visual Basic

- F9 will toggle a breakpoint
- F5 will resume normal program execution to the next breakpoint

■ Stepping through a procedure

- F8 will manually control line-by-line execution
- Use Shift F8 to step over a procedure in a macro

■ Adding a watch expression while in break mode

- View intermediate values of counters in loops and other variables
- Useful for finding logic errors

Programming

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Debugging Tips

■ Indent your code for readability

- easier to trace and decipher indented code
- indented coded enhances code documentation

■ Turn on syntax checking

- Tools | Options
- General module tab - check Display Syntax errors

■ Require variable declarations (Option Explicit)

■ Break down complex procedures into many small chunks

■ Enter all macros using lowercase

- if Visual Basic does not change a keyword to normal case, then you typed the keyword incorrectly

Programming

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Debugging Tips

■ When a macro refuses to run

- Make sure the module containing the local macro is open
- If you are trying to run the macro using a shortcut key, be certain the shortcut key has been defined
- Check to see whether multiple macros were assigned the same shortcut key. If so, change one of the procedures
- Make sure that another open module doesn't have a procedure with the same name

■ Use comments to temporarily deactivate parts of code that are giving you problems

■ Do not use reserved words and worksheet names for macro

■ Use range names =percent_raise rather than =B7

Programming

OPIM 101

 **harton**
The Wharton School
of the University of Pennsylvania

Debugging Tips

■ Use lowercase "variable" names to tell variables from reserved keywords

■ Command macros do not need a return value

■ Function macros must return an argument

■ Break up long statements

- difficult to debug complex formulas and procedures
- reduce to smaller chunks to facilitate isolation of the problem

■ Use user-defined constants

- if constants are always constant, declaring values as constants
 - prevents values from changing
 - makes the code easier to understand
 - prevents you from using the wrong value in a formula

Programming

OPIM 101

 **harton**
The Wharton School
of the University of Pennsylvania

THE JOURNAL

OF THE

AMERICAN

SOCIETY

OF THE

SCIENCE

OF THE

ARTS

AND

LITERATURE

OF THE

CITY

OF

NEW YORK

1851

1852

1853

1854

1855

1856

1857

1858

1859

1860

1861

1862

1863

1864

1865

1866

1867

1868

1869

1870

1871

1872

1873

1874

1875

1876

1877

1878

1879

1880

1881

1882

1883

1884

1885

1886

1887

1888

1889

1890

1891

1892

1893

1894

1895

1896

1897

1898

1899

1900

1901

1902

1903

1904

1905

1906

1907

1908

1909

1910

1911

1912

1913

1914

1915

1916

1917

1918

Simulation

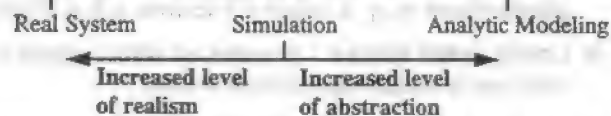
- What is simulation?
- Selected applications of simulation
- Comparing analytic solutions with simulation
- Advantages and disadvantages of simulation
- Types of simulation
- Random number generation
- Discrete event simulation logic
- Statistical analysis of simulation output
- Simulation languages and animation

Simulation
OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

What is simulation?

One model of reality at varying levels of detail



- Uses a computer program that duplicates the essential behavior of a real physical system
- Inputs are given to the simulation program
- Outputs are computed by the simulation program
- Systematic manipulation of the inputs allows the evaluation of alternative decision policies
- Select most desirable alternative from all possible runs
- No guarantee that an optimum solution will be found

Simulation
OPIM 101

 **Wharton**
The Wharton School
of the University of Pennsylvania

Advantages of simulation

- **Realism** - simplifying assumptions for analytical models are not needed. Any degree of complexity can be handled.
- **Time compression** - months of real time can be simulated in seconds on the computer (e.g. weather forecasts)
- **Training** - simulation requires less mathematical training than analytical modeling
- **Presentation of results** - results are often easier and more intuitive to understand, especially if animation is used to "see" a proposed system in operation
- **Impossible to solve analytically**
- **Actual observation or operation is too expensive or takes too much time**

Simulation

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Disadvantages of simulation

- **Failure to optimize** - tyranny of large combinations of alternatives (e.g. 3 labor x 4 looms x 8 staffing levels = 96)
- **Long lead times** - months of effort might be required to develop a major simulation model
- **Lack of generality of results** - results only apply to situation in the model and can not be extended out of context. Must understand the underlying process to start.
- **Costs for developing simulation capability** - hardware, software, training, support, and staffing
- **Model must contain uncertainty** - otherwise the solution will be deterministic
- **Misuse of simulation** - because it is easier to build simulations, people who are not fully qualified can build models that are incorrect or incomplete.

Simulation

OPIM 101

 **Wharton**

*The Wharton School
of the University of Pennsylvania*

Types of simulation

■ Risk analysis models

- Probabilistic bounds on ROI to quantify degree of uncertainty

■ Monte-Carlo models

- Process model with little knowledge of parameter bounds
™ Crystal Ball

■ Time-based simulation

- continuous time simulation (e.g. oil refinery processes).
- discrete event simulation (e.g. bank teller staffing).
 - static models show steady state equilibrium
 - dynamic model show transient response of the system to changes in inputs

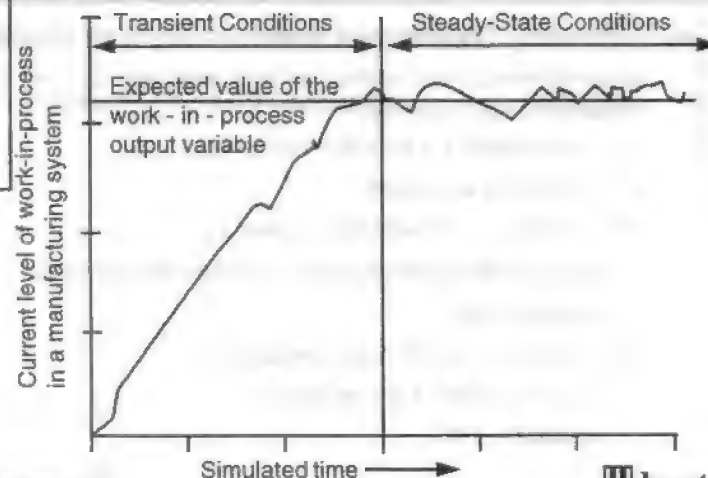
Simulation

OPIM 101



The Wharton School
of the University of Pennsylvania

Transient vs. Steady State Operation



Simulation

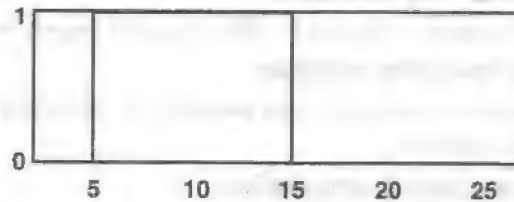
OPIM 101



The Wharton School
of the University of Pennsylvania

Random number generation

■ Uniform distribution over the range 5 to 15



$$r_1 = 5 + 10 * \text{rand}()$$

$$= 5 \text{ when } \text{rand}() \text{ is } 0$$

$$r_2 = 5 + 10 * \text{rand}()$$

$$= 14.9999 \text{ when } \text{rand}() \text{ is } .9999$$

■ To include both 5 and 15 use:

$$r_3 = 5 + \text{int}(10 * \text{rand}() + .1)$$

$$= 15 \text{ when } \text{rand}() \text{ is } .9999$$

note: $\text{int}()$ returns the integer portion of a number

Simulation

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Random number generation

■ Uniformly distributed pseudo-random numbers

A common method of random number generation is to use a multiplicative congruential formula $R_n = k \times R_{n-1} \text{ (modulo 1)}$

$R_0 = .0123456789$ (the initial seed value)

$k = 100,003$ (a constant)

$R_1 = 100,003 \times .0123456789 \text{ (modulo 1)}$
 $= 1234.6049270367 \text{ (modulo 1) returns the remainder}$
 $= .6049270367$

$R_2 = 100,003 \times .6049270367 \text{ (modulo 1)}$
 $= 60,494.5184511101 \text{ (modulo 1)}$
 $= .5184511101$

Simulation

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

EXCEL Random number generation

RAND() Returns an evenly distributed random number greater than or equal to 0 and less than 1. A new random number is returned every time the worksheet is calculated.

Remarks To generate a random real number between a and b, use:

RAND() * (b - a) + a

If you want to use RAND to generate a random number but don't want the numbers to change every time the cell is calculated, you can enter =RAND() in the formula bar and press F9 (or COMMAND + = in Microsoft Excel for the Macintosh) to change the formula to a random number.

Example To generate a random number greater than or equal to 0 but less than 100: RAND() * 100

In Visual Basic use RND()

Simulation

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Discrete event simulation logic

■ Cumulative frequency functions

Interarrival time in hours (time between arriving ships)	Cumulative frequency with which these times occur
0 to 6	0.1
6 to 12	0.2
12 to 18	0.9
18 to 24	1.0

If random number = 0.7414,
what is the interarrival time (IAT) ?



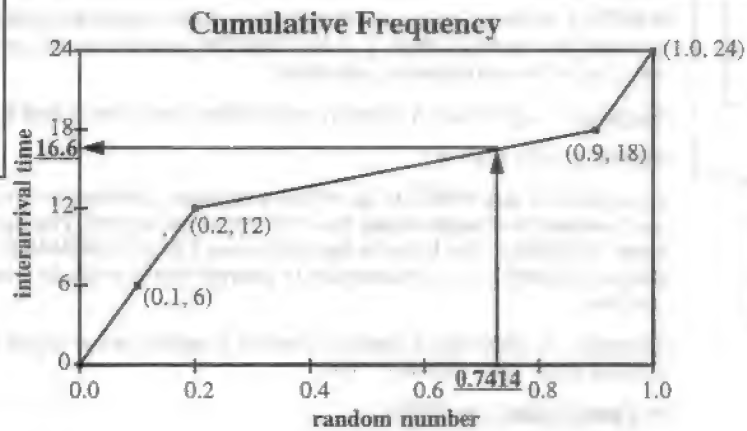
Simulation

OPIM 101

Wharton

The Wharton School
of the University of Pennsylvania

Discrete event simulation logic



Simulation

OPIM 101

 The Wharton School
 of the University of Pennsylvania

Discrete event simulation logic

■ Cumulative frequency functions

Within any interval of the distribution, values of the function are uniformly distributed. Interpret cumulative frequency functions as a series of straight-line segments that connect the function defined ordered pairs of points.

Interarrival time in hours (time between arriving ships)	Cumulative frequency with which these times occur
0 to 6	0.1
6 to 12	0.2
12 to 18	0.9
18 to 24	1.0

$$IAT_0 = IAT_1 + (rand_0 - cfreq_1) \times ((IAT_2 - IAT_1) / (cfreq_2 - cfreq_1))$$

if $rand_0 = 0.7414$

$$= 12 + (0.7414 - .2) \times ((18 - 12) / (.9 - .2))$$

$$= 16.6$$

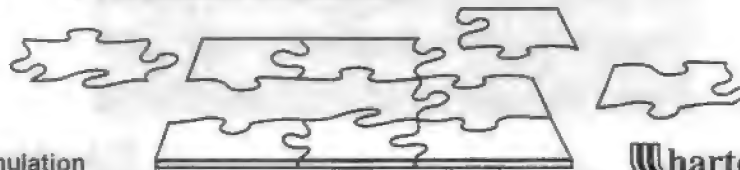
Simulation

OPIM 101

 The Wharton School
 of the University of Pennsylvania

Discrete event simulation logic

- Maintain "clock" to keep track of events
- Process events in time sequence
- Determine next type of event
- Process according to real world "event logic"
- Update statistics describing the simulation
- Terminate simulation

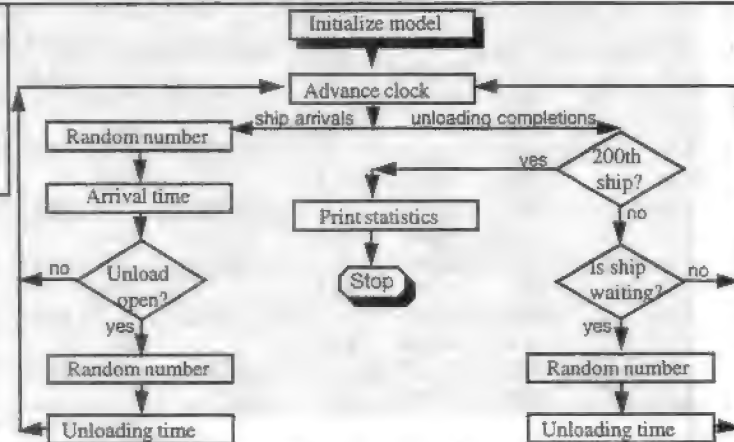


Simulation

OPIM 101

 The Wharton School
 of the University of Pennsylvania

Harbor Simulation (Textbook page 82-84)



Simulation

OPIM 101

 The Wharton School
 of the University of Pennsylvania

Harbor Simulation (Textbook page 82-84)

Order Drawn	Random Number	0-24 hrs IAT	12-16 hrs UNLOAD
1	0.2068	12.1	12.8
2	0.7295	16.5	14.9
3	0.3441	13.2	13.4
4	0.5436	14.9	14.2
5	0.3091	12.9	13.2
6	0.2276	12.2	12.9
7	0.4833	14.4	13.9
8	0.7414	16.6	15.0
9	0.7666	16.9	15.1
10	0.0273	1.6	12.1
11	0.0700	4.2	12.3
12	0.6489	15.8	14.6
13	0.9564	21.4	15.8
14	0.2389	12.3	13.0

Simulation

OPIM 101

 **Wharton**
*The Wharton School
of the University of Pennsylvania*

Harbor Simulation (Textbook page 82-84)

Que	No.	Clock	Event	Schedule				Wait
				Arrival at	Begin Unload	IAT	Unload	
0	0	0.0	Initialize	12.1		12.1	14.9	
0	1	12.1	Arrive	25.3	12.1	13.2		27.0 0.0
0	2	25.3	Arrive	40.2		14.9		
1	1	27.0	Depart		27.0		13.2	40.2 1.7
0	3	40.2	Arrive	52.4		12.2		
0	2	40.2	Depart		40.2		13.9	54.1 0.0
1	4	52.4	Arrive	69.0		16.6		
0	3	54.1	Depart		54.1		15.1	69.2 1.7
1	5	69.1	Arrive	70.7		1.6		
0	4	69.2	Depart		69.2		12.3	81.4 0.1
1	6	70.6	Arrive	86.4		15.8		
0	5	81.5	Depart		81.5		15.8	97.3 10.8
1	7	86.4	Arrive	98.7		12.3		
Total waiting time								14.3
Average waiting time								2.4

Simulation

OPIM 101

 **Wharton**
*The Wharton School
of the University of Pennsylvania*

Statistical analysis of simulation output

- Trade-off between number of berths and waiting time
- Utilization of berth capacity
- Maximum number of ships waiting
- Mean waiting time (23.69 hours transient only)
- Std Dev (14.36 hours transient only)
- 95% Confidence interval vs point estimate

$$23.69 - 1.96 \times (14.36 / 100^{.5}) = 22.33$$

$$23.69 + 1.96 \times (14.36 / 100^{.5}) = 25.05$$

Simulation
OPIM 101



Wharton
The Wharton School
of the University of Pennsylvania

Statistical analysis of simulation output

For a sample mean \bar{x} and a sample standard deviation s we can construct a **confidence interval** within which we can be reasonably sure contains the true population mean.

$$95\% \text{ confidence interval for } \mu = \bar{x} \pm 1.96 \frac{s}{\sqrt{n}}$$

Of course, 5% of the time the population mean will fall outside this interval. This is true because the sample means are normally distributed and 5% of the values of a random variable in a normal distribution fall more than 2 standard deviations from the mean.

Simulation
OPIM 101

Wharton
The Wharton School
of the University of Pennsylvania

Statistical analysis of simulation output

■ iid Assumptions

- independent replications using unique random numbers
- identical distribution of random numbers (e.g., still uniform from 12 to 16 but the specific random values change)

■ Central Limit Theorem

- no matter what distribution is followed by the point estimates, their average will be approximately normally distributed for samples of size 30 or more. This allows us to compute a confidence interval using a z-statistic rather than a t-statistic.

■ How many replications? What batch size?

- depends on computer time and size of model
- importance of the decision and if needed in "real time"
- quadrupling the sample size halves the confidence interval

Simulation

OPIM 101



The Wharton School
of the University of Pennsylvania

Simulation languages and animation

```
STUDENT GPSS/H RELEASE 2.0 (AY130) FILE: ds101b.gps
SIMULATE          base time unit: 1 minute
ATM STORAGE 1     define number of ATM machines
GENERATE 2,2       people arrive, one by one
QUEUE LINE        start LINE queue membership
ENTER ATM         request/capture the ATM machine
DEPART LINE       end LINE queue membership
ADVANCE 3,1       conduct an ATM transaction
LEAVE ATM         done with the ATM transaction
TERMINATE 0       leave the system

GENERATE 480       minutes per 8 hours
TERMINATE 1       end simulation
START 1           simulate 480 minutes of service
END              end of Model-File execution
```

Simulation

OPIM 101



The Wharton School
of the University of Pennsylvania

Simulation languages and animation

ATM SIMULATION OUTPUT

Simulation begins. RELATIVE CLOCK: 480 minutes

ATM UTILIZATION	QUEUE CONTENTS
163 ENTRIES	227 TOTAL ENTRIES
1 MAXIMUM CONTENTS	64 MAXIMUM CONTENTS
2.938 AVERAGE SERVICE	31.595 AVERAGE LINE
1 CAPACITY CONTENTS	66.809 AVERAGE TIME/UNIT
1 CURRENT CONTENTS	64 CURRENT CONTENTS
0.998 AVERAGE UTILIZATION	

Are these statistics realistic?

Simulation

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

Simulation languages and animation

■ Proof Animation

- post-simulation tool
- language independent (portable)
- vector-based and file-driven
- includes CAD-like tools and CAD file import/export

■ Demonstration

■ Animation provides

- visual program feedback for debugging
- enhanced presentation convinces audience of model realism
- time-consuming to develop
- might be examined in lieu of rigorous statistical analysis
intuitions from the animation could be incorrect

Simulation

OPIM 101

Wharton

*The Wharton School
of the University of Pennsylvania*

**OPIM 101 Introduction to the Computer as an Analysis Tool**

Fall 1995

Operations and Information Management Department

Final Exam

The Wharton School of the University of Pennsylvania

Professor Lohse

The University of Pennsylvania Academic Code of Integrity states:

Any work that a student undertakes as part of progress toward a degree or certification must be the student's own, unless the relevant instructor specifies otherwise. That work may include examinations, whether oral or written, oral presentations, laboratory exercises, papers, reports, and other written assignments.

This two hour exam is closed book and closed notes. All answers must be written neatly or printed carefully on the exam. No credit will be given when handwriting can not be read. You may separate the pages for convenience in working the exam. Use the back of the pages for scratch paper. We have a stapler you can use to refasten the pages when you turn in your exam. Write your name and section number on the top of each exam page. Misconduct during an examination is a violation of the Code. Any student who violates this Code will receive a zero for the work in question and will be referred to the Judicial Inquiry Officer for further action.

I agree to abide by the provisions of the Code of Academic Integrity, and I certify that I will comply with the Code in taking this examination.

Signature

Print your name _____ Social Security Number _____

Circle your section number

12:00
section 1

1:00
section 2

2:00
section 3

GOOD LUCK and Season's Greetings from the staff

Exam Section	Points Available	Your Points
Multiple Choice	20	
Visual Basic Programming	7	
Linear Programming	20	
Decision Analysis	20	
Database	27	
Simulation	6	
Total	100	

OPIM101-FALL95 name _____ section number _____

12. Steady-state conditions in a simulation,
- A contain the only observations that should be used for statistical analysis.
 - ☐ B vary stochastically around some expected value.
 - C are independent and easily discernible from transient conditions.
 - D none of the above
13. Bob Jones is the database administrator for Budget Airlines. Marketing staff are complaining to him that reservation data for confirmed passengers exclude two passengers on every Boeing 727 plane. This database issue is most related to
- A data independence.
 - ☐ B data integrity.
 - C data dictionary.
 - D data security.
14. All of the following are disadvantages of neural networks except
- A The actual rules embodied in the neural network are not readily apparent
 - ☐ B The neural network can not self-optimize to automatically adjust their parameters to learn over time.
 - C Neural networks will not work well at solving problems for which sufficiently large and general sets of training data are not available.
 - D Complex, expensive computers with multiple processors are needed for large complex problems.
15. A linear program for maximizing total contribution with an objective function that can be made infinitely large without violating any of the constraints is called
- A redundant.
 - ☐ B unbounded.
 - C extreme.
 - D infeasible.
16. A survey of every student in OPIM 101 finds that 70% feel they will earn a grade of "A". An heuristic or human information processing bias that best accounts for these survey results is
- A group think
 - ☐ B overconfidence
 - C illusion of control
 - D availability

OPIM101-FALL95 name _____ section number _____

17. Which of the following is not a valid URL?

- A http://opim.wharton.upenn.edu
- B ftp://guru.cern.ch
- C gopher://world.std.com:70/
- D html://library.princeton.edu

18. A neural network used for stock price prediction would most likely

- A be an unsupervised neural network.
- B be a supervised neural network.
- C be a feedforward neural network.
- D none of the above.

19. Which of the following techniques is used to find the optimum solution for a certain type of large, complex, problems?

- A linear programming
- B neural networks
- C discrete-event simulation
- D genetic algorithms

20. All the of the following are reasons for using simulation models *except*

- A for evaluation of the fitness of individuals in a genetic algorithm.
- B for models without uncertainty.
- C when real world measurement and observation of an existing system is too disruptive or expensive.
- D when the underlying assumptions for analytical models are not met.

OPIM101-FALL95 name _____ section number _____

(7) Visual Basic Programming and graphs

Sub question1() 'What value is displayed in the message box for countj and counti

Dim i, j, countj, counti As Integer '1 point each

counti = 0

countj = 0

For j = 100 To 20 Step -3

countj = countj + 1

For i = -30 To 0 Step 2

counti = counti + 1

Next i

Next j

MsgBox "The value of count is " & countj, 0, "For outer loop" ' _____ 27

MsgBox "The value of count is " & counti, 0, "For inner Loop"

End Sub ' _____ 432

Sub question2() 'What is the label of the message box displayed by this macro?

Dim a, b, c, d As Integer '2 points

Dim flag As Boolean ' _____ Message Box 1

a = 1

b = 2

c = 3

d = 4

flag = False

If (((a + d) = (c + b) And Not (flag)) Or a = b) Then

MsgBox "False", 0, "Message Box 1"

Else

MsgBox "True", 0, "Message Box 2"

End If

End Sub

Sub question3() 'What is the value of *total* displayed in the message box ?

Dim total, sum(3), max As Integer '2 points

max = 3 ' _____ 55

sum(1) = 20

sum(2) = 10

sum(3) = 5

total = question(sum, max)

MsgBox "Total equals " & total, 0, "What is the value of total?"

End Sub

Function question(sum, max)

Dim i As Integer

For i = 1 To max

question = question + sum(i) * i

Next i

End Function

OPIM101-FALL95 name _____ section number _____

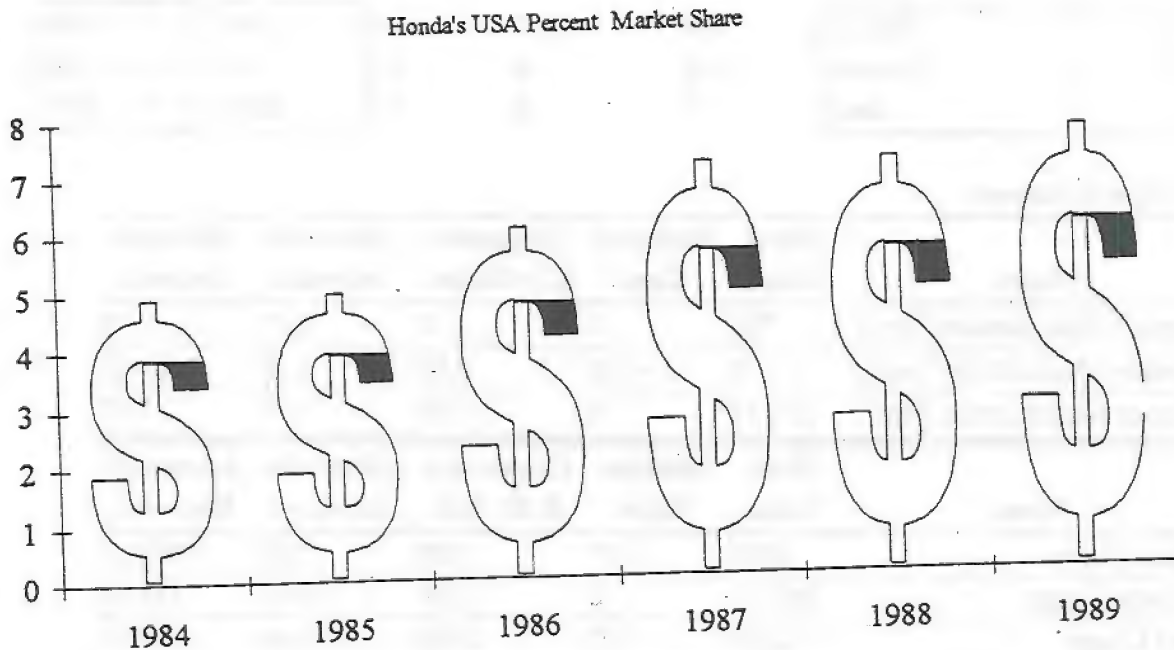
(1) Graphs

The graph below will be presented in a senior level management meeting to discuss marketing strategies for Honda. What is the graph trying to show?

Slope of the linear increase in market share over time.

Comment on the appropriateness of this graph to display the data accurately.

The dollar signs are difficult to compare and identify a trend. 2D value displays impart no information in second dimension. No label for percentage on y axis. Shaded portion at tip of dollar sign confuses viewer as to where the data values are.



OPIM101-FALL95 name _____ section number _____

(20) **Linear programming.** Parker Sisters manufactures ball-point pens, mechanical pencils and fountain pens. The company is trying to plan its production mix for each week. Joe believes that the company can sell any number of pens and pencils it produces, but production is limited. Because of a recent strike and certain cash-flow problems, the suppliers are only willing to deliver at most 1000 ounces of plastic, 1200 ounces of chrome, and 2000 ounces of stainless steel each week. These are variable costs since suppliers do not require Parker Sisters to buy a fixed amount each week. Use the model below and the Excel Solver output to answer the following questions. (Assume each question is independent unless otherwise stated).

	A	B	C	D	E	F	G
1	Parker Sisters Inc.	Ballpoint	Mechanical	Fountain			
2		Pen B	Pencil	Pen F	Totals		
3	Number to manufacture	700.00	0.00	133.33	833.33		
4	Total Contribution	\$ 3.00	\$ 3.00	\$ 5.00	\$ 2,766.67		
5							
6					Usage	Max	
7	Plastic	1.2	1.7	1.2	1000.00	<=	1000
8	Chrome	0.8	0	2.3	866.67	<=	1200
9	Steel	2	3	4.5	2000.00	<=	2000

Sensitivity Report

Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
Number Manufacture Pen B	700	0	3.00	2.00	0.78
Number Manufacture Pencil	0	-1.38	3.00	1.38	1E+30
Number Manufacture Pen F	133.33	0	5.00	1.75	2.00

Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
Plastic Usage	1000	1.17	1000	200	466.67
Chrome Usage	866.67	0	1200	1E+30	333.33
Steel Usage	2000	0.8	2000	555.56	333.33

a. Name all of the constraints that are binding. Explicitly state each type of constraint!

(1 point each; 3 points maximum)

Plastic usage (structural)

Steel usage (structural)

Number Manufacture Pencil ≥ 0 (non-negativity)

b. A local distributor has offered to sell Parker Sisters an additional 500 ounces of stainless steel for \$0.60 per ounce more than it ordinarily pays. Should the company buy the steel at this price? Support your answer by explaining what happens to Parker Sisters' product mix. What is the total weekly contribution if it does buy the stainless steel?

(1 point shadow price comparison; 1 points mix; 1 point contribution; 3 points maximum)

Yes (\$0.60 < \$0.80) Parker Sisters is willing to pay up to an \$0.80 premium for stainless steel. $500 \times \$0.80 = \400 increase in total weekly contribution (\$2,766.67 + \$400 = \$3,166.67). The mix will change!

OPIM101-FALL95 name _____ section number _____

- c. What must the minimum contribution margin per mechanical pencil be in order to make them worthwhile to produce? Explain your response.

(2 points; no partial credit)

At a contribution margin greater than $\$3.00 + \1.38 ($\$4.38$), objective ranging is no longer valid. Mechanical pencils may become part of the mix at this point.

- d. Parker Sisters buys its plastic for $\$5.00$ per ounce. This week, Parker Sisters has an opportunity to sell 300 ounces of plastic to another company for $\$6.50$ per ounce. The other company does not produce pencils or pens and is not a competitor. Should Parker Sisters sell the plastic? Support your answer by calculating the change in total weekly contribution if Parker Sisters' does sell the plastic?

(1 point for each part; 4 points maximum)

- (1) The $\$1.50$ premium for plastic ($\$6.50 - \5.00) is greater than the marginal value to Parker Sisters (the shadow price is only $\$1.17$). A 300 ounce decrease is within the allowable range for the shadow price.

(1) $300 \times \$1.17 = \351 decrease in total weekly contribution ($\$2,766.67 - \$351 = \$2,415.67$)

(1) PLUS a $300 \times \$1.50$ increase from the sale of plastic ($\$450 + \$2,415.67 = \$2,865.67$)

(1) The net increase in total weekly contribution is $\$99.00$ per week!

- e. The R&D department at Parker Sisters has been redesigning the mechanical pencil. The new design requires 1.1 ounces of plastic, 2.0 ounces of chrome, and 2.0 ounces of stainless steel. If the company can sell one of these pencils with a contribution margin of $\$3.00$, should it approve the new design? Explain your response.

(0.5 points each correct marginal value; 0.5 points for 2.887; 1 point for explanation; 3 points maximum)

$1.1 \text{ ounces plastic} \times \$1.17 + 2 \text{ ounces chrome} \times \$0 + 2.0 \text{ ounces steel} \times \$0.80 =$

$(1.287 + 0 + 1.60) = \$2.887$

$\$2.89 < \3.00 Contribution is greater than marginal cost therefore YES approve!

- f. Marketing believes that the company should produce at least 20 mechanical pencils per week to round out its product line. What effect would this have on total weekly contribution margin? Explain your response.

(1 point partial credit for calculation; 1 point for explanation; 2 points maximum)

Reduced cost is $\$1.38$ per pencil $\times 20$ pencils = $\$27.60$ decrease per week

$(\$2,766.67 - \$27.60 = \$2,739.07)$

- g. If the per-unit contribution margin per ball-point pen decreases to $\$2.25$, what is the new total weekly contribution margin and the new product mix? Explain your response.

(1 point partial credit for calculation; 1 point for explanation; 2 points maximum)

$\$2.25$ is within the allowable range ($\$3.00 - 0.78$ allowable decrease).

The mix does not change! 700 Ball-point Pen and 133.33 Fountain Pens

$700 \times \$0.75 = \525 decrease in total weekly contribution ($\$2,766.67 - \$525 = \$2,241.67$)

- h. The chrome supplier might have to fulfill an emergency order, and would be able to send only 1000 ounces of chrome this week instead of the usual 1200 ounces. What effect would this have on total weekly contribution margin? Explain your response.

(1 point; no partial credit)

No change! They only use 866.67 ounces per week & shadow price is $\$0$ for chrome

OPIM101-FALL95 name _____ section number _____

(20) **Decision Analysis.** Colonial Motors is trying to determine what size of manufacturing plant to build for a new car it is developing. Only two plant sizes are under consideration: large and small. The cost of constructing a large plant is \$25 million and the cost of constructing a small plant is \$15 million. Colonial Motors believes a 70% chance exists that the demand for this new car will be high and a 30% chance that it will be low. The following table summarizes expected profits as a function of factory size and demand.

Colonial Motors can purchase a national marketing survey from Megabucks Consulting that will measure consumer attitudes towards the new car. Consumer attitudes can be favorable or unfavorable. If they are favorable, demand will be high. If they are unfavorable, demand will be low. The survey will cost \$0.250 million. When demand is high, they successfully predict a favorable consumer attitude 6/7 of the time. When demand is low, they successfully predict a unfavorable consumer attitude 7/9 of the time. The following table summarizes these conditional probabilities.

Conditional probabilities for a given level of demand (high or low)

Expected Profits (in millions) if		
Factory Size	High Demand	Low Demand
Large	\$175	\$95
Small	\$125	\$105

Survey	High	Low
Favorable	6/7	2/9
Unfavorable	1/7	7/9

Enter and calculate the probabilities given in the table below:					
		Prior probability	Conditional probability	Joint probability	Posterior probability
Survey	High Demand	.7	6/7	.6000	.90
Favorable	Low Demand	.3	2/9	.0667	.10
Survey	High Demand	.7	1/7	.1000	.30
Unfavorable	Low Demand	.3	7/9	.2333	.70

The table is worth 4 points. 16 entries at .25 points each.

What is the probability that the survey finds a favorable consumer attitude? _____ 2/3 (0.5 points)

What is the probability that the survey finds a unfavorable consumer attitude? _____ 1/3 (0.5 points)

Complete the decision tree on the next page. Label all branches. Neatly show all calculations.

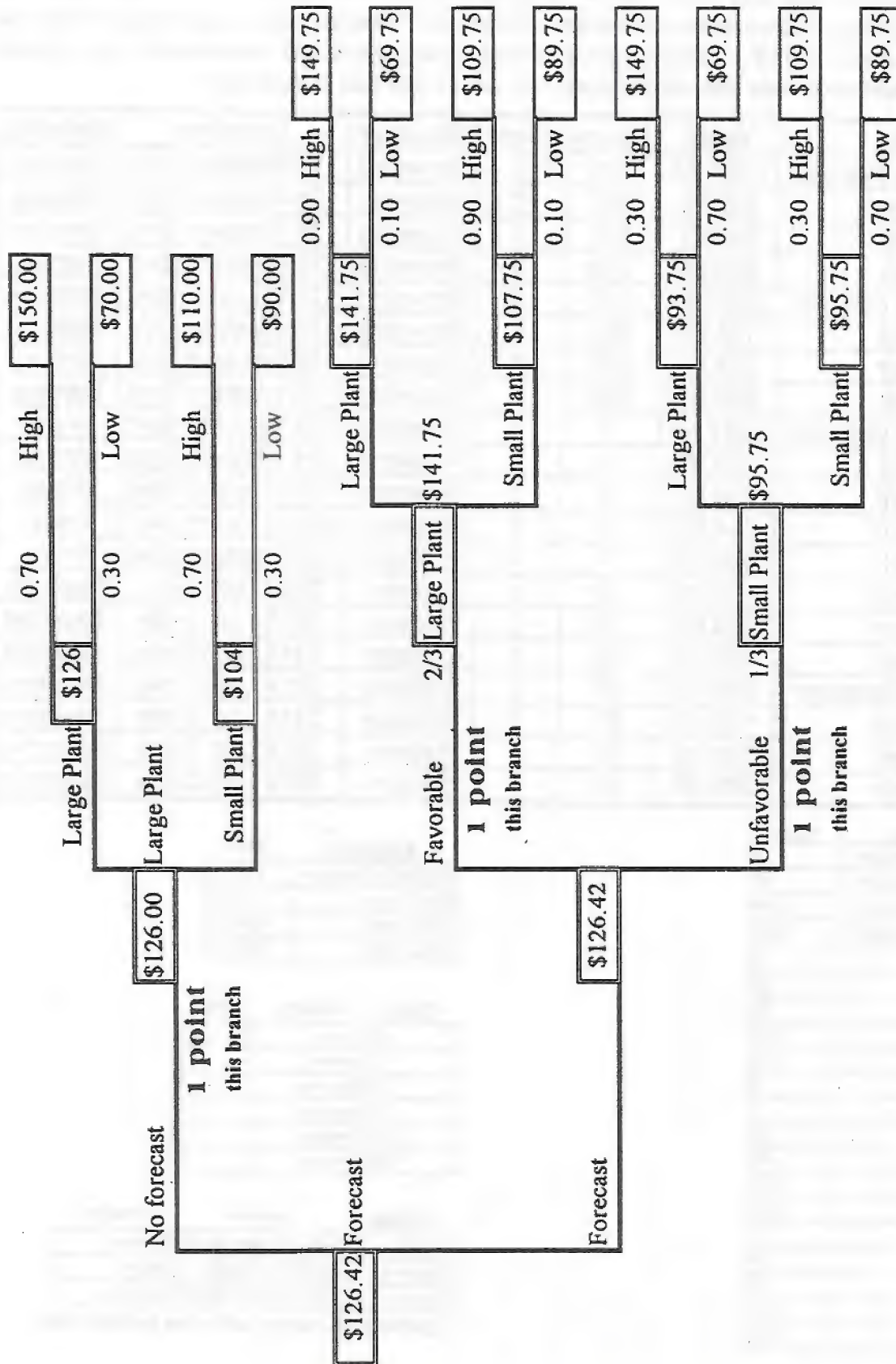
EMV without sample information _____ \$126 million (2 points; no partial credit)

EMV with sample information _____ \$126.42 million (2 points; no partial credit)

EPPI _____ \$132 million (2 points; no partial credit)

EVPI _____ \$6 million (3 points; no partial credit)

EVSI _____ \$666,667 (3 points; no partial credit)



(10) **Database Normalization** The inventory tracking system for Lotek Industries identifies the location of the corporation's office furniture and computers. The only table in the "database" contains the following data shown below. Assume that this data is the entire universe of records. (A) Using proper normalization procedures, convert the "database" into a collection of relational tables that are all in 3NF. Label the primary keys, foreign keys, and concatenated keys on each table. To save time, only state the relations. Do not list the data in each table.

Equipment	Value	Room	Floor	Manager	Location	Building
486DX2 Computer	2367.93	219	2	Browning	126 Western Blvd	Ramsey
Floor Lamp	123.99	219	2	Browning	126 Western Blvd	Ramsey
Office chair	234.82	219	2	Browning	126 Western Blvd	Ramsey
Office desk	989.06	219	2	Browning	126 Western Blvd	Ramsey
486DX2 Computer	2367.93	303	3	Browning	126 Western Blvd	Ramsey
Floor Lamp	123.99	303	3	Browning	126 Western Blvd	Ramsey
Office chair	234.82	303	3	Browning	126 Western Blvd	Ramsey
Office desk	989.06	303	3	Browning	126 Western Blvd	Ramsey
486DX2 Computer	2367.93	5	1	Rowland	124 Western Blvd	Annex
Floor Lamp	123.99	5	1	Rowland	124 Western Blvd	Annex
Office chair	234.82	5	1	Rowland	124 Western Blvd	Annex
Office chair	234.82	7	1	Rowland	124 Western Blvd	Annex
Office desk	989.06	7	1	Rowland	124 Western Blvd	Annex
Floor Lamp	123.99	23	1	Bellamy	2318 Lod Circle	Bayonne
Office chair	234.82	23	1	Bellamy	2318 Lod Circle	Bayonne
Office desk	989.06	23	1	Bellamy	2318 Lod Circle	Bayonne
486DX2 Computer	2367.93	87	1	Bellamy	2318 Lod Circle	Bayonne
Floor Lamp	123.99	87	1	Bellamy	2318 Lod Circle	Bayonne
Office chair	234.82	87	1	Bellamy	2318 Lod Circle	Bayonne
Office desk	989.06	87	1	Bellamy	2318 Lod Circle	Bayonne

Equipment	Room
486DX2 Computer	5
486DX2 Computer	87
486DX2 Computer	219
486DX2 Computer	303
Floor Lamp	5
Floor Lamp	23
Floor Lamp	87
Floor Lamp	219
Floor Lamp	303
Office chair	5
Office chair	7
Office chair	23
Office chair	87
Office chair	219
Office chair	303
Office desk	7
Office desk	23
Office desk	87
Office desk	219
Office desk	303

Equipment	Value
486DX2 Computer	2367.93
Floor Lamp	123.99
Office chair	234.82
Office desk	989.06

Room	Building	Floor
5	Annex	1
7	Annex	1
23	Bayonne	1
87	Bayonne	1
219	Ramsey	2
303	Ramsey	3

Building	Location	Manager
Annex	124 Western Blvd	Rowland
Bayonne	2318 Lod Circle	Bellamy
Ramsey	126 Western Blvd	Browning

2 points each correct table - no partial credit

OPIM101-FALL95 name _____ section number _____

(B) Suppose the following record is added to the universe of records. Does the collection of relational tables in third normal form created in (A) above change? If so, what changes would you make? You may not change any relation. Be certain to list the revised table or tables containing this record with all of the data from the data in part (A) above.

Office chair	134.28	7	1	Rowland	124 Western Blvd	Annex
--------------	--------	---	---	---------	------------------	-------

<u>Equipment</u>	<u>Value</u>
486DX2 Computer	2367.9
Floor Lamp	123.99
Office chair	234.82
Office chair	134.28
Office desk	989.06

Add new entry into Equipment value table.
Equipment and Value must be composite primary key!

1 point - no partial credit

(C) Suppose the following record is added to the universe of records (*in addition to the one record in (B) above*). There are now three chairs in room 7. Two are priced at \$234.82 and one is priced at \$134.28. Does the collection of relational tables in third normal form created in (A) above change? If so, what changes would you make?

Office chair	234.82	7	1	Rowland	124 Western Blvd	Annex
--------------	--------	---	---	---------	------------------	-------

Equipment and Value no longer uniquely identify an inventory item. Must add a new attribute that will uniquely key each entry in this table
(e.g. Equipment identification number).

1 point - no partial credit

OPIM101-FALL95 name _____ section number _____

(3) Database Transitive Dependencies The following table contains student advising information. Students status is based on the following classification. Freshman have 30 credit hours or less. Sophomores have 60 credit hours or less. Juniors have 90 credit hours or less. Seniors have 91 credit hours or more. An advisor (AdvLname) only advises in his/her major teaching field. For example, a Computer Information Systems (CIS) professor only advises CIS majors, and English professor only advises English majors, and so on. Each advisor has his/her own office (AdvOffice). Advisors do not share offices. Each office has a unique identification. For example, there is only one office identified as KOG-109. The only table in the "database" contains the following data shown below. Assume that this data is the entire universe of records.

StuNum	StuMajor	StuHours	StuClass	AdvLname	AdvOffice
10025	Management	87	Jr	Gonzales	KOG-209
10026	Marketing	15	Fr	Robertson	KOG-328
10027	English	48	So	Nealy	VLG-164
10028	Fine Arts	37	So	Kallenberger	FA-234H
10029	Management	93	Sr	Gonzales	KOG-209
10030	CIS	72	Jr	Rafferty	CSB-320
10031	English	105	Sr	Nealy	VLG-164
10032	CIS	114	Sr	Nealy	CSB-453
10033	Mathematics	93	Sr	Hanoon	MGR-125
10034	Geography	102	Sr	Williams	HUH-205
10035	English	45	So	Antons	VLG-202

(A) Make a list of all transitive dependencies in the "database".

StuClass is a calculated field with a transitive dependency on StuHours

AdvOffice has a transitive dependency on AdvLname - knowledge of office identifies the advisor

AdvLname IS NOT DEPENDENT on StuMajor. Look at Nealy. Advisors with the same last name teach and advise in different academic areas. If there was a unique faculty ID number, then an advisor major transitive dependency would exist.

StuMajor IS NOT DEPENDENT on AdvLname. Two advisors have the same last name. Nealy and Antons both advise in English.

A transitive dependency exists when an attribute is dependent on another attribute that is neither a primary key nor part of a composite primary key

(B) Suppose a student record is deleted from the table. What deletion anomalies may occur? Could delete information about all advisors for a majors if only 1 advisor/major

Information about the faculty advisor and his/her office is lost.

OPIM101-FALL95 name _____ section number _____

Job Relation

JOB_CODE	JOB_DESCRIPTOR	JOB_CHG
1	Support	\$32.50
2	Engineering	\$56.00
3	Database Admin.	\$49.90

Division Relation

DIV_CODE	DIV_DESCRIPTOR
1	Marketing
2	Research
3	Production
4	Info. Systems

Qualification Relation

QUAL_CODE	QUAL_NAME
1	Bachelor's degree
2	Master's degree
3	Pd.D degree
4	Technical Certification
5	Non-tech. Certification

Degree Relation

EMP_CODE	QUAL_CODE
104	1
104	5
105	1
107	1
107	2
107	3
108	1
108	2
108	3
110	1
112	1
112	4
113	1
113	5
114	1
114	2
114	4
115	4

Dependents Relation

EMP_CODE	DEP_NUM	DEP_DOB
105	1	09/15/70
105	2	02/12/89
105	3	12/23/93
107	1	11/14/89
108	1	04/05/65
108	2	10/16/79
111	1	12/17/65
111	2	12/30/93
112	1	09/02/80
114	1	01/29/68
114	2	06/22/87

Employee Relation

EMP_CODE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	JOB_CODE	EMP_PHONE	DIV_CODE
104	Leondas	Ekiri	U	3	456-1234	3
105	Holmes	George	H	1	456-3390	1
106	Washington	Theresa	T	1	456-3389	4
107	Smith	Jerome		2	514-2216	2
108	Roberts	Anne	M	2	821-6675	2
109	Stovall	Peter	G		249-8816	
110	Steinbauer	Jeanne	M	1	514-9007	1
111	Colby	Georgette	W		514-0181	1
112	Koczisko	Paul	K	3	456-2556	4
113	Rosenthal	Herman	R	1	514-1128	1
114	Williamson	Mary	G	2	456-2887	3
115	Rockwell	William		1	574-9113	3

OPIM101-FALL95 name _____ section number _____

(6) Conceptual Database Design

- a) How many Access files (*.mdb) would be used to store these tables? _____ 1
- b) How many tuples does *Degree* contain? _____ 18
- c) How many attributes does *Employee* contain? _____ 7
- d) How many fields does *Employee* contain? _____ 7
- e) Identify the primary and foreign keys for each of the following tables. If there is no foreign key, enter NONE under the foreign key heading. If a table has a concatenated key, identify all components.

4 points total; 16 answers @ .25 points each

Table	Primary Key	Foreign Key
Employee	EMP_CODE	JOB_CODE, DIV_CODE
Qualification	QUAL_CODE	NONE
Job	JOB_CODE	NONE
Degree	EMP_CODE + QUAL_CODE	EMP_CODE, QUAL_CODE
Division	DIV_CODE	NONE
Dependents	EMP_CODE + DEP_NUM	EMP_CODE

(4) SQL Query Part A Write an SQL query that returns a frequency table showing the count of the number of employees having a Bachelor's, Master's, or Ph.D. degree. QUAL_NAME and Count of QUAL_CODE are the two attributes in the dynaset.

Microsoft Access 2.0 SQL (1point partial credit for each clause; 3 points maximum)

```
SELECT DISTINCTROW QUALIFICATION.QUAL_NAME, Count(DEGREE.QUAL_CODE) AS CountOfQUAL_CODE
FROM QUALIFICATION INNER JOIN DEGREE ON QUALIFICATION.QUAL_CODE = DEGREE.QUAL_CODE
WHERE ((DEGREE.QUAL_CODE <= "3"))
GROUP BY QUALIFICATION.QUAL_NAME;
```

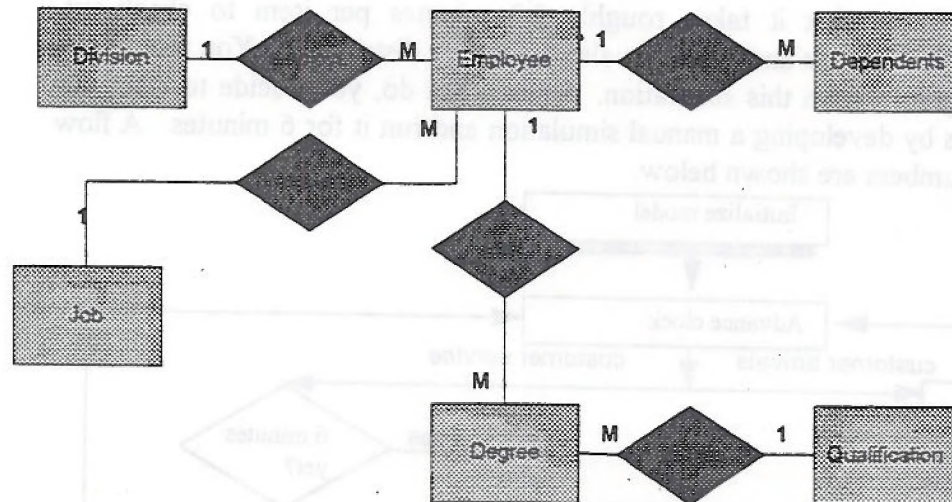
Primis Book SQL (1point partial credit for each clause; 3 points maximum)

```
SELECT DISTINCTROW QUAL_NAME, Count(QUAL_CODE) AS CountOfQUAL_CODE
FROM QUAL
WHERE QUAL_CODE <= "3"
GROUP BY QUAL_NAME
```

SQL Query Part B What will the resulting SQL query in Part A return? Show this dynaset below.

QUAL_NAME	CountOfEMP_CODE
Bachelor's degree	8
Master's degree	3
Ph.D degree	2

- (4) **ER Diagram** Draw a complete entity-relationship diagram. Include all entities, and relationships. Name the relationships and label (1:1, 1:M, and M:M). Do not show any attributes.

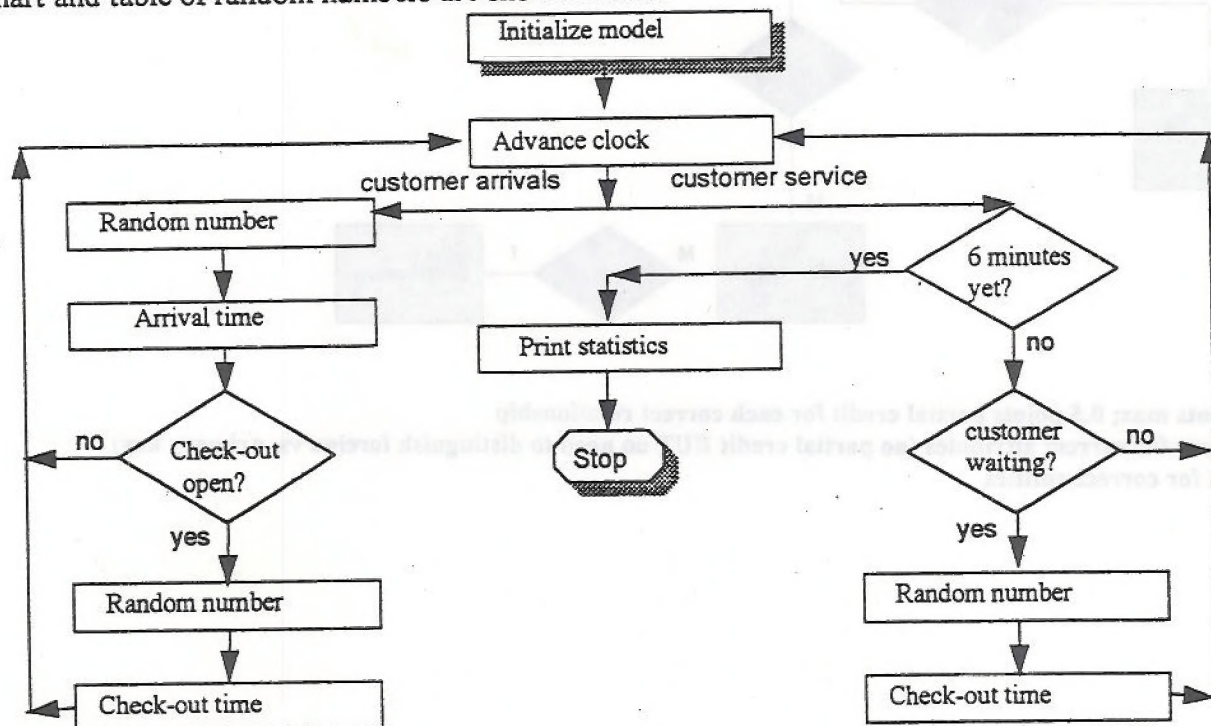


2.5 points max; 0.5 points partial credit for each correct relationship

0.5 points for correct attributes (no partial credit BUT no need to distinguish foreign vs. primary key)

1 point for correct entities

(6) **Simulation** A small convenience grocery store has one check-out counter. Currently, customers form a single queue and wait in line. The store is very concerned about peak rush demand and they are considering adding another check-out counter. Before making this investment, the store owner decides to develop a simulation model of the check-out line. An industrial engineer studied the patterns regarding how much time lapses before a new customer arrives in the line during the noon lunch hour. Customers arrive in line every two minutes. The engineer concludes that this time is uniformly distribution with a range from 1 minute to 3 minutes. Further, it was found that it takes roughly 0.3 minutes per item to check-out. Customers purchase from 1 to 10 items and these are also uniformly distributed. You would like to build a Visual Basic program to run this simulation. Before you do, you decide to test your understanding of the process by developing a manual simulation and run it for 6 minutes. A flow chart and table of random numbers are shown below.



random_number(1)	=	0.2068
random_number(2)	=	0.9295
random_number(3)	=	0.0441
random_number(4)	=	0.1436
random_number(5)	=	0.8081
random_number(6)	=	0.2833
random_number(7)	=	0.6276
random_number(8)	=	0.5414
random_number(9)	=	0.7666
random_number(10)	=	0.0273
random_number(11)	=	0.0700
random_number(12)	=	0.6489
random_number(13)	=	0.9564
random_number(14)	=	0.2389

Be certain that you adhere to standard simulation modeling practices. Any arrival into a model will automatically generate a successor. In the event of time ties, you must free the server, then update the flow of transactions in the model. Initialize the model with first arrival and service time.

Use the table on the following page to record the movement of transactions through the simulation. Print neatly! Show all work. Use effective labels. Carry 3 significant digits after the decimal point in your final answers.

Random Number	What happens at this clock time?	Clock	Current Arrive Number	Interarrival Time	Arrival Time	Current Depart Number	Time Required to Check-out	Depart Time
0.2068	Initialize	0.0	1	1.41	1.4	0	0	0
0.9295	Initialize	0.0	1	1.41	1.4	1	2.81	4.2
0.0441	Arrive	1.4	2	1.09	2.5	1	2.81	4.2
0.1436	Arrive	2.5	3	1.29	3.8	1	2.81	4.2
0.8081	Arrive	3.8	4	2.62	6.4	1	2.81	4.2
0.2833	Depart	4.2	4	2.62	6.4	2	1.06	5.3
0.6276	Depart	5.3	4	2.62	6.4	3	1.99	7.3
0.5414	Terminate	6.0	4	2.62	6.4	3	1.99	7.3
0.7666								
0.0273								
0.0700								
0.6489								
0.9564								
0.2389								

Fill out the table so that only one random number is used per row

Total Waiting Time	3.932	Total NumberWaiting	3	Average Waiting Time	1.311
--------------------	-------	---------------------	---	----------------------	-------

Do you think this answer is "correct"? Give at least two reasons.

At least 2 reasons @ 0.5 points each Transient conditions vs steady-state Number of replications per run Statistical summary of data

